University of Cape Town

Performance Analysis of Hybrid WiFi and TV White Space Links

by

Mokwape Magdeline Lamola

A dissertation submitted in fulfilment of the requirements for the degree of Master of Science

in the
Faculty of Science
Department of Computer Science

Supervisors:
Dr. David Lloyd Johnson, University of Cape Town
Dr. Melissa Densmore, University of Cape Town
Dr. Albert Lysko, Council for Scientific and Industrial Research

Abstract

Internet access has the potential to improve economic growth in developing countries, yet in developing countries with emerging economies, such as South Africa, Internet access opportunities are not evenly distributed. This digital divide exists between urban and rural areas and even within urban areas in many developing countries. Urban areas are densely populated - simplifying telecommunication infrastructure roll-out, whereas rural areas are sparsely populated - making the roll-out of telecommunication infrastructure considerably more complex and expensive. This digital divide poses a significant challenge since a large portion of the developing country's population is based in rural areas.

Cellular, satellite and some pockets of WiFi technologies are mostly used to provide access in rural areas. Although these technologies help mitigate connectivity challenges in rural areas, they are often costly and provide limited broadband access. The high cost of access in rural areas is due to the lack of fibre for backhaul that provides cost effective bulk wholesale capacity and the use of costly satellite links or cellular links for Internet access. Cost-effective technology alternatives such as WiFi and/or Television White Space (TVWS) can provide an effective approach to provide affordable last mile and middle mile connectivity for Internet access in many of these poorly connected areas.

TVWS provides excellent coverage and penetration through vegetation, buildings since it utilizes spectrum in the UHF bands currently used for Television broadcasting to offer broadband wireless connectivity. Although TVWS has good propagation characteristics in some non-line-of-sight (NLOS) scenarios and can offer better coverage than WiFi, thanks to the mass production and massive industry and development support behind it, WiFi provides low-cost connectivity with better throughput speeds in line-of-sight (LOS) scenarios. Previous research has focused on the characteristics and performance of TVWS and WiFi in isolation. This study aims to describe how their individual characteristics can then be used to compliment each other for improvement in the last-mile access.

This work looked at the performance of WiFi and TVWS technology in different settings, including line-of-sight, non-line-of-sight environments and using different combinations of these technologies. Experiments focused on the performance of WiFi (IEEE 802.11a) and TVWS (IEEE 802.11g cards downconverted to UHF) with an objective to help improve connectivity in areas with poor coverage, due to environmental factors, such as vegetation and distance. The study utilized the Council of Scientific and Industrial Research's (CSIR's) Meraka Institute custom built White Space Mesh Node (WSMN) equipped with WiFi and TVWS radio interface cards to carry out the experiments. The study particularly focuses on the 5 GHz Wi-Fi and Ultra High Frequency (UHF) 530 to 600 MHz frequency bands.

The study presents an analysis of data collected over the dual-radio wireless network in indoor and outdoor environments. Presentation of this data follows measurements of single radio and aggregate radio link traffic collected in various line-of-sight and non-line-of-sight environments. These measurements deduce the effects of environment on 5 GHz and TVWS frequency band,

effects of modifying performance parameters, improvement or degradation of aggregated TVWS WiFi links, and the usage of the measured performance data for network planning.

Each experiment tests different combinations of radio settings, such as channel, transmit power and channel width to measure throughput, signal strength, packet loss, and Signal to Noise Ratio. These tests were done in both indoor and outdoor environments. The results collected and presented in this work show that although TVWS has superior propagation characteristics compared to WiFi, its performance is often poorer than WiFi when there is clear line-of-site and at shorter distances. The study, in addition, presents data that shows that the overall radio performance in a network is affected by more than just spectrum availability in space or time, but also by radio settings and the environment. The study also goes on to show that aggregated links, that combine both TVWS and WiFi, do not always lead to better network performance. The study lastly presents tailored scenarios of single and aggregated radio links that lead to better performance with the hope that these will help network designers and researchers make better-informed decisions on how to use available radio resources effectively.

Contents

List of Figures

List of Tables

List of Abbreviations

WiFi	Wireless Fidelity
TVWS	Television White Space
UHF	Ultra High Frequency
VHF	Very High Frequency
NLOS	Non-Line-Of-Sight
LOS	Line-Of-Sight
SU	Secondary User
QoS	Quality of Service
WLAN	Wireless Local Area Network
IEEE	Institute of Electrical and Electronic Engineers
LAN	Local Area Network
MAN	Metropolitan Area Network
WRAN	Wireless Regional Area Network
ICASA	Independent Communications Authority of South Africa
OSPFv2	Open Shortest Path First version 2
WiMAX	Worldwide Interoperability for Microwave Access
SNR	Signal-to-Noise-Ratio
CA	Carrier Aggregation
ISM	Industrial, Scientific and Medical
LTE	Long-Term Evolution
DL	Down-Link
UL	Up-Link
UPS	Uninterruptible Power Supply
WSMN	White Space Mesh Node
HPN	High Performance Node
ICT4D	Information and Communications Technology for Development
GPS	Global Positioning System
CSV	Comma-Separated Values

DTV	Digital Television
BATMAN	Better Approach To Mobile Adhoc Network
RTT	Round-Trip Time
TCP	Transmission Control Protocol
WSM	Wireless Mesh Network
WNIC	Wireless Network Interface Controller

CHAPTER 1

OVERVIEW OF THE STUDY

1.1 Introduction

This research study analyses various performance challenges experienced in a dual 5 GHz WiFi and UHF (470 MHz - 694 MHz) Television white spaces radio network in developing regions, and suggests solutions to improve the performance of these dual radio systems. This chapter presents the background, problem statement, hypothesis, and significance of the study.

1.2 Background

Statistics South Africa (Stats SA) and Internet World Stats reported that the South African population had an Internet penetration of approximately 52% in 2015 that increased by at least 7% in the year 2016 [34]. A greater part of that population was located in urban areas. There is a significant difference between urban and rural areas with regards to Internet availability. Average mobile Internet access in South Africa in 2017 was 39.6% for rural residences and 61.5% for urban residences. According to the general household survey (GHS) published by the Stats SA in January 2016, only one-tenth of South African households have access to Internet at home [34]. Independent Communications Authority of South Africa (ICASA) reports a 92.8% national population for 4G/LTE in 2019.With Northern Cape reporting the lowest LTE coverage at 72% in 2019 and Western Cape reporting an 82% LTE coverage rate [18]. Providing Internet access to rural areas is difficult for a variety of reasons. Some common challenges are high network infrastructure deployment expenses and the high cost of mobile communications for users. Due to the challenges rolling out fixed-line access in rural areas, the primary medium of Internet access is through mobile network operators.

According to [9], 26.8% of individuals in rural areas access Internet via mobile phones and 46.85% in urban areas in the year 2016. Affordability, though, remains a major barrier for low-income areas when accessing the Internet using mobile phones. ICT Analytical research by the Association for Progressive Communications shows that the communication services are not affordable in many regions of South Africa. On average, individuals in rural communities depend on government social grants. The social grants range from R 380.00-R 1,620.00 per month and a significant number of those users sacrifice much of this income on limited mobile services [4]. Although the affordability is dependent on individual income,the cost of airtime is standardized nationally with low-income users ,on prepaid packages, paying a much higher premium for services than wealthier contract users. Individuals often have to sacrifice up to 25% of their income on mobile voice and data. South Africa continues to lack meaningful growth in affordable Internet access that provide alternatives to costly cellular services. This growth will be critical for making the country regionally and globally competitive.

This research focuses on the provision of low-cost Internet access and connectivity in low-income, rural or poorly serviced areas using affordable WiFi and TV White Space technology.

In achieving the underlying idea of providing cheaper and efficient Internet access, we integrate already existing network technologies. Recent research has been generated from the need to use licensed spectrum to provide alternative affordable Internet access, where the spectrum is accessed dynamically allowing users to utilize it when not in use by licensed users (services), with licensed users entitled to protection. The frequencies that are reserved for TV broadcasting are located in the UHF and VHF bands. The idea is to dynamically access the spectrum as a secondary user to opportunistically reuse the given bandwidth, known as Television White Spaces (TVWS). Research shows TVWS to have complementary features to WiFi that are most commonly used in the 2.4 GHz and 5 GHz frequency bands. The crucial issue when building networks using these technologies is understanding the characteristics and performance that each technology offers and which radio technology is most suited to the unique terrain or interference environment at hand.

Some of the technologies explored to achieve connectivity in rural or under-serviced areas are: license-free WiFi, satellite and cellular technology. Readily available satellites and mobile operators offer costly Internet access, whereas WiFi provides a more cost-effective alternative. Deployments of license free WiFi are inexpensive, due to the low cost of equipment and no need for costly spectrum licenses. Despite this advantage of WiFi, the technology requires a clear line of sight in order to function in a satisfactory manner.

Numerous research studies have looked at improving some of the shortcomings of WiFi. These include longer range connectivity, better performance in non-line-of-site scenarios, increasing the speed and performance of the radio technology, and good or better interference avoidance compared to traditional technologies. Some new technologies and techniques that have been explored are TVWS and newer WiFi protocols. Recent research reveals the need to use the licensed television broadcasting spectrum to alternatively provide affordable Internet access in rural areas. To achieve this, the spectrum reserved for Television broadcasting located in the UHF and VHF bands is accessed dynamically by allowing users to utilize the spectrum when not in use by licensed users (primary users). The idea is to dynamically access the underused spectrum as a secondary user to opportunistically reuse the given bands for broadband known as TVWS.

Ongoing research that has been built around TVWS has a resonating interest in this field of study given the current increasing demand for connectivity. The traditional stagnant allocation of its spectrum has led to the under-utilization.

In considering the aforementioned technologies, TVWS can interchangeably outperform WiFi and vice versa. TVWS performs at a greater coverage level while WiFi covers approximately 100 m in indoor scenarios and under good conditions, i.e., clear Fresnel zone. Low-density secondary users (SUs) give an advantageous performance. TV bands have a long wavelength covering large portions of an area and can penetrate through vegetation and challenging terrain [36]. Although these technologies are different, they can each deliver the complimentary performance in best-suited conditions. Currently, the WiFi spectrum is often mostly utilized and subject to a high degree of interference in urban areas [11]. These technologies can each deliver excellent performance in unique conditions. In some scenarios integrating both technologies proved to be the best solution. To improve the quality of service (QoS), ultimately merging their complementary characteristics in a radio network.

IEEE 802.11

WiFi, which stands for Wireless Fidelity, is a technology used for wireless local area networks (WLAN) [29]. The technology spans through corners of the world and is one of the most widely used media for connecting to the Internet. WiFi creates links between devices supporting the IEEE 802.11 standard. This is associated with the Institute of Electrical Engineers standardization body (IEEE) [27]. WiFi is associated with the IEEE 802 standard, that envelops the local and metropolitan area networks (LAN and MAN). The specific standard that encompasses WiFi relates to IEEE 802.11 a/b/g/n. Devices communicating within this standard, communicate in 900 MHz;

Protocol	**Frequency (GHz)**	**Bandwdith (MHz)**	**Range**		**Speed (Mbits/s)**	**Compatibility**
			Indoor	Outdoor		
802.11 a	5 & 3.7	20	35 m	120 m	54	N/A
802.11 b	2.4	20	35 m	140 m	11	802.11 (DSSS)
802.11 g	2.4	20	38 m	140 m	54	802.11 (HR-DSSS)
802.11 n	2.4 & 2.5	20 & 40	70 m	250 m	100	802.11 a/b/g

Table 1.1: IEEE 802.11 PHY Standards: Revised from [29]"Wireless Networking in The Developing World", page 128

2.4 GHz; 3.6 GHz; 5 GHz and (or) 60 GHz frequency bands. For the purpose of this study, focus is directed towards the 2.4 and 5 GHz frequency bands. With the 2.4 GHz band in the 802.11 (b, g, n) standards and 5 GHz band under the 802.11 (a, n, ac) standard.

IEEE 802.22

This standard is related to the Wireless Regulation Area Networks (WRAN). The devices operating in this standard are known to utilize the white spaces existing in the television (TV) frequency spectrum. IEEE 802.22 standard allows the usage of unused spectrum, originally allocated for television broadcasting services, and used on a secondary basis. This spectrum is then used for the wireless network and this technology is known as TVWS [29].

1.3 Problem Statement

Internet access is considered a good factor with the ability to improve economic growth in a country, wherein in other countries, it is considered a "human right". It poses good benefits in improving universal competitiveness of countries, especially developing countries. However there still remain challenges regarding universal internet accessibility, the major challenge being the need to extend internet access beyond urban areas to rural areas. This challenge remains critical as the majority of the population in developing countries is based in rural and/or remote areas. Although the Internet is accessible in other areas, it still remains expensive. In this case, the use of inexpensive or low-cost and widely available wireless technology will help mitigate this challenge. Technologies such as commercial satellite, mobile operators and license-exempt technologies, namely WiFi. However internet access offered by satellite and operators does not necessarily help bridge the low-cost gap, the deployments of these technologies are usually expensive with limited access. Which leaves license-exempt WiFi as a better candidate to help solve this challenge [1], although WiFi is readily available and low-cost to deploy it is widely adapted for longer coverage[35]. Deploying or providing connectivity services in urban areas where there is high population density, is prevalent and inexpensive. Therefore it is easy for internet service providers to roll out services in these areas [40]. It is conversely expensive in other geographic regions. And these are small remote areas where users are widely scattered. Although WiFi has proven to be a good way to access broadband, the deployments of it predominantly favors urban or metropolitan areas, which support high Internet speeds [16][2]. This technology, although attractive for this challenge, has its flaws and is not ideal for all geographical regions. WiFi works relatively well at short distances in clear line-of-sights [41], [8]. However, TVWS, compared to WiFi, performs at greater distances and non-line-of-sight. This is to specify setups with faint clear view, mountainous areas and even highly vegetated areas [41].TVWS is more ideal for sparsely populated and rural areas because it uses low-frequency radio [20]. Both WiFi and TVWS have their own merits in mitigating the problem, and the choice between the two is not obvious. The use of either WiFi or TVWS depends on a number of factors such as the presence of interference from external or internal factors; the height of the network setup; the distance between the nodes. This then proves the need to examine and understand the behavior of WiFi and TVWS. Understanding factors such as "what affects license-exempt WiFi and TVWS?", "how does network parameters such as channel?", "channel width and transmit power affect the overall performance of the network using these two wireless technologies?" and "lastly how can the characteristics of each wireless technology be used to improve the overall performance of the network?". In understanding these questions this study will improve future network deployments and designs on which wireless technology to use.

1.4 Hypothesis of the Study

In honoring the proposal of this study this subsection defines three hypotheses stated as follows:

1. **The use of unlicensed or license-exempt technologies together with TVWS can help overcome Internet access in developing countries.**
   Previous research has made arguments on the use of TV White Spaces to overcome the digital divide. That research also includes countries such as South Africa, where there is a struggle in Internet access distribution. This study proves more relevant given the recent regulations on the use of television white spaces prescribed by the Independent Communications Authority of South Africa (ICASA) in the year 2018 [3]. Therefore this study will be prominent for future communications society.

2. **TVWS and WiFi technologies do not necessarily apply to all network deployments.**
   Although the use of TVWS and license-exempt spectrum to improve Internet access in developing countries, the technologies possess characteristics prone to external constraints. Therefore different deployment scenarios pose unique constraints. In order to deploy these technologies, there is a need to understand the behavior of each technology. This is important considering the unique possible deployment scenarios in different environments.

3. **Improving radio link quality, complementing radio characteristics.** The use of both TVWS and WiFi has no clear performance distinctions between the two. However, there exists different characteristics in both these radios, characteristics that complement each. TVWS offers long range coverage, ensuring high penetrative attributes and WiFi owning it merits in offering a better performance in short range clear sites. The use of these differences to improve the quality of the network is highly beneficial.

1.5 Significance of the Study

This work is a subset that contributes to a project run in Ocean View in the Western Cape province. The project is called *"Inethi"*, and aimed at bridging the digital divide in this community. This concept explains how there is a gap between different communities where access to needed information and communication technology is concerned. The project's main objective is the use of down converted WiFi in order to expand the information in this developing community in the project 802.11, a/b/g WiFi and down-converted WiFi into the UHF band, also known as TV White Spaces. However, implementing such a solution in Ocean View would require taking environmental factors into consideration. Ocean View consists of different geographical topology. Nodes are positioned at different locations in the communities, such as schools, taxi ranks, and libraries. Therefore this work plays a role in analyzing the performance of both these radios in order to investigate how TVWS and WiFi performances could be weighed against each other in different topology. Topology , in this case, refers to how spatially separated the nodes are positioned and obstructions between nodes (such as trees, buildings or mountains). With this analysis, this study will have a clearer idea of how TVWS and WiFi differ with different topology, and thus make informed decisions on how and when to improve links in the network and also how to make informed decision on deploying nodes in the township.

1.6 Concepts and Definitions

To understand and characterize the quality of a network, this subsection presents measurement-based terms and concepts that help interpret the usage of these terms and understand the context in which they are referenced in this document.

Fresnel Zone(s):

This Fresnel Zone theory is named after the physicist Augustin-Jean Fresnel. It helps define how much of the space around a transmitting and receiving antenna is affected in a propagation path. We consider the line of propagation from the transmitting and receiving node and space around those nodes to contribute the overall received signal. Therefore, according to microwave characteristics waves travel directly from the transmitter to the receiver or also bounce off and is reflected to the receiver. This therefore affects the received signal and then contribute to the way the network behaves. Although the theory considers the existence of more than one Fresnel zones, the first Fresnel zone is the main zone of concern due to the imperative effects it has on the propagating signal from the transmitter to the receiver. With that being said, the first Fresnel zone should at least be kept clear of obstructions. This theory then suggests that at least 60% of the radius of the first Fresnel zone should be clear. This suggests the following formula to achieve the radius:

$$r = 17.31\sqrt{\frac{d_1 * d_2}{f * d}}$$

OR

$$r = \sqrt{\frac{\lambda * d_1 * d_2}{d}}$$

where:
r = radius of the zone (m)
(d_1 *and* d_2) = distances from the obstacle to the link end point (m)
d = total link distance (m)

Line-of-sight and Near-Line of sight (Non-line-of-sight) propagation:

Line of sight is the term used in radio communication as radio transmission path that is clear and has no obstruction between the transmitter and receiver.

Non-line-of-sight is a term referring to a propagation path being obstructed by any obstacle, thus not providing a clear visual line of sight between the source and receiver. Obstacles include:

- Buildings
- Mountains
- Trees
- Weather (rain, wind)

1.7 Publications

The work presented in this study represent an extensive academic journey that features work that contributes to the the development of ideas presented in this study. Below the study presents publications relating to the overall discussion in this study.

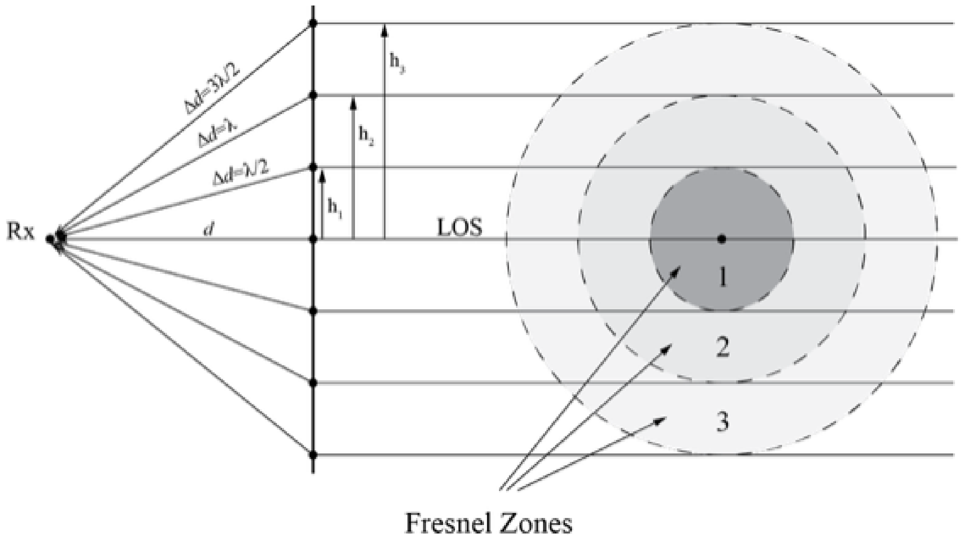


Figure 1.1: The description of Fresnel-Zones. Note: Reprinted from "Line of Sight Obstruction Revision:10/16", by Campbell Scientific, Inc, 2016, App. Note Code: 3RF-E,Copyright 2016 by Campbell Scientific Inc.

- **TVWS devices Spectrum Mask Test and Analysis** 2pp- (*SATNAC: 4-7 September 2016*)
  This work presents experimental analysis of emitted signal and spectrum mask of low-cost TVWS devices at various channels with different power budgets. The experiments test the permissible power level at different secondary channels utilized in the TV broadcasting spectrum by Doodle lab TVWS device. The work in this paper contributes methods on how to improve improve communication design [21].

- **Head to head battle of TV White Space and WiFi for connecting developing regions** 10pp- (*8th EAI International Conference on e-Infrastructure and e-Services for Developing Countries (Africomm) 6-7 December 2016*)
  This paper presents experimental performances of 5 GHz WiFi links and TV White space links using down-converted WiFi. The work carries out 5 GHz WiFi and TV White Space band experiments in line-of-sight links and obstructed links.The paper show that various network parameters and environment affects the WiFi and TVWS links. Therefore, concluding on 5 GHz WiFi to have the best link performance where scenarios are short range line-of-sight and outperforms TVWS and TVWS outperforms the best NLOS scenarios compared to WiFi[19].

- **Experimental Analysis of 5 GHz WiFi and UHF-TVWS hybrid Wireless Mesh-** (International Conference on Cognitive Radio Oriented Wireless Networks: CROWNCOM 2018: Cognitive Radio Oriented Wireless Networks pp 3-14)
  This paper investigates link performances over 5 GHz WiFi and UHF-TVWS hybrid backhaul and show possible link permutation to optimize the overall network. The paper proposes a link selection model. The model is based on analytic hierarchy process and grey relational analysis. Link selection model is investigated in a multi-band multi-radio wireless mesh nodes [22].

- **Performance Analysis of Dual 5 GHz WiFi and UHF TV White Space Network Links (Best Paper Award)-** (*IEEE Wireless Africa Conference (WAC) 2019*)
  Commonly used WiFi is known to be ill-suited for penetrating vegetation and buildings and non-line-of-sight conditions. Television white space (TVWS) operates in ultra-high frequency (UHF) bands that overcome many of the penetration and line-of-sight challenges found in the 2.4 GHz and 5 GHz bands normally used by WiFi. The aim of this study is to report on the performance of WiFi technology in the 5 GHz band and the TVWS technology in the 600 MHz UHF TV band as well as a combination of both radios in two different scenarios, short-range clear line-of-sight, and non-line-of-sight conditions. A number of performance metrics are compared for varied distances and increasing levels of vegetation in the propagation path. A measurement script collects the estimated throughput, bitrate, signal strength, noise, transmit power, transmit error, packet loss, and round trip time. Both TVWS and WiFi Experiments showed increased sensitivity to noise as channel widths increased with TVWS being particularly susceptible to noise in nearby channels from powerful TV transmitters. Aggregating the WiFi and TVWS radios proved to have the best performance improvements when the WiFi and TVWS links had similar throughput in line-of-sight conditions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to understand the radios used in this experimental study, a few characteristics need to be addressed and studied. For the past decade, there has been research focusing on the performance of wireless networks, and even considers ways of improving performances of these networks based on the behavior of connectivity and environment.

2.2 Radio waves propagation

It is essential to study the propagation of radio waves and characteristics of electronic devices for the purpose of understanding how radio communication works. Radio waves, known as electromagnetic waves, travel between network points of communication generated by electronic equipment. The communication between two points or nodes is assisted by antennas that help radiate the fed energy from electronic equipment through free space. When energy spreads out from one antenna, known as a transmitting antenna, it is then absorbed by the receiving antenna.

The energy that travels from a transmitter to a receiver is referred to as a *signal*. So ideally this signal can travel freely in a vacuum without disturbance, but once a medium is introduced in the propagation path the signal then gets reflected; refracted; scattered; diffracted or absorbed.

2.3 Behavioral performances of WiFi

WiFi provides a relatively inexpensive alternative to the costly options of providing Internet access. The main disadvantages of WiFi are that it has a short transmission range and requires clear line-of-sight (LOS). Studies have been carried out to measure the propagation factors in WiFi [17], [7], [30] in order to measure the performances in the presence of obstructions. The propagation measurements of an indoor and outdoor scenario is performed in a 2.4 GHz band and shows that the performance of WiFi is degraded in a non-line-of-sight. Another study that highlights WiFi performing poorly in the presence of obstructions, tested the 2.4 GHz and 5 GHz interfaces and found that the morphology of an environment affects the radio signal propagation [7]. The general outcomes point out obstructions and environmental topology such as vegetation, height, distance increase path loss and signal attenuation or even inducing jitter [30]. Depending on the focus of these experimental outcomes, other literature experimentally evaluates the maximum range of which WiFi can cover. This includes experiments in LOS and NLOS links in different environments. Following the work in [35] where the authors study WiFi-based communication between nodes in a tunnel, this substantiates the argument the literature is making. Experimental evaluation of WiFi averages the best performance of WiFi to reach at least 150 m range at clear line of sight and a maximum of 225 meters range non-line of sight.

2.4 Behavioral performances of TVWS

In view of the limitations of WiFi highlighted in the previous section, a lot of attention has grown around using TV bands on a secondary basis by accessing the unused spectrum. Cognitive radio (CR) [26] technology is used to help access other licensed TV bands (white spaces), known as TVWS. According to the Meraka Institute in South Africa, there is proof of abundant spectrum in rural areas [23] giving a foundation to the work that follows in this project. To measure the compatibility of each channel to be used, parameters such as the throughput, latency, and SNR are generally analyzed. Following such a route is research undertaken in Malawi, showing measured throughput and latency being affected by the distance from one current node to the base station [24]. With the results measured over a 7.5 km link, the measured results show an average SNR of 24.7 dB, data-rate of 420 kbps and a latency of 118 ms. The results are followed up by a performance of the network, where the performances are analyzed in a rural secondary school in Malawi [25] confirming a coverage range of 7.5 km and resulting in a maximum throughput of 2 Mb/s. The performance of the WiFi-like secondary network in an indoor and outdoor setting is reviewed in [33] and analyzed to show that the performance of TVWS is not as ideal in more realistic conditions. The authors prove the operation of WiFi access points (APs) in TVWS is attractive in outdoor rural areas due to low user demand in such areas. TV white space devices have an interference problem with nodes on the same network and those from heterogeneous networks [32].

2.5 Combined Radios

Both WiFi and TVWS have advantageous characteristics to perform optimally or even outperform each other in a network. There are benefits in using both radios or even the combination of each radio with other available wireless radios. A study conducted between 5 GHz WiFi and TVWS found that WiFi performs better than TVWS in a short ranged distance of at least 500 m, and TVWS shows a better performance rate than WIFi in longer ranges of at least 2-2.2 km. The differences in WiFi and TVWS are studied and analyzed in the 5 GHz and TVWS bands in the Western Cape [19] using channel 36 (5180 MHz) for WiFi and 575 MHz frequency in TVWS to measure 500 m short range and 2-2.2 km long range Line-of-sight and Non-line-of-sight experiments. The work shows WiFi to perform better than TVWS in a short range distance with a throughput 1.7 Mbps higher than TVWS, and TVWS performs better than WiFi in longer ranges. In the presence of a tree as an obstruction, TVWS has a throughput of 5.28 Mbps as compared to WiFi. Their standard parameter measures were taken in these researches to weigh any prevailing strengths in the network. Similarly, WhiteMesh technology is used in assigning channels in a joint WiFi and White space bands in a backhaul tier of a wireless mesh network [10]. The study uses population density as the primary parameter for channel availability and usability. In any case, WiMAX also played a slightly similar role as that of TVWS, and the need to collaborate their features was explored [11]. In incorporating low cost, flexible and heterogeneous network the system capacity of the network is improved. To evaluate the performance of each of the six algorithms they measured parameters such as throughput, jitter, and an end to end delay in the IEEE 802.16e and 802.11n technologies. The authors [11] concluded that OSPFv2 being the best option in integrating WiFi and WiMAX, having low jitter, a low end-to-end delay, high throughput and high-speed mobile users. Alternately, with the introduction of OSPF for IPv6 (OSPFv3) similar output could be achieved with the use of OSPFv3. Although, OSPFv2 has shown work of integration between two radios and the translated possibility of OSPFv3 achieving similar if not more optiins, the inclusion of this concept is outside the scope defined for this work. Ultimately, both radios have desirable features, and in other work, the benefits of WiMAX and WiFi are integrated. They explore the idea of bandwidth sharing and channel collaboration services are integrated focusing on channel competition and optimal WiFi coverage for the model [28]. The decisions in this paper are based on the rate of customer demand concerning the availability, the number of users it can accommodate and the cost of the different service providers. Making propositions to meet real-time traffic demands by raising the amount of capacity.

Other research following up on attenuation induced by vegetation or even buildings have used those characteristics to their advantage. More specifically the idea is utilized in a radar system for target or shielding purposes. Radar uses vegetation as a potential improvement in other systems, radar UHF and microwave frequencies are studied by Gómez-pérez and Cuiñas to model the attenuation and scattering induced by vegetation [14]. The study categorizes the vegetation species into seven categories to evaluate the performance of the radar invisibility cover demonstrating that the probability of detection in the radar system is stable in higher frequency bands, and the denser, the vegetation foliage, the higher the SNR reduction.

Although a handful of research has been done on the use of WiFi in TVWS or with other radio technologies, the work focuses significantly on the benefits of using one radio over the other according to coverage advantages of TVWS over WiFi. Other interests are built around the aggregation of Wi-Fi like secondary systems in TVWS in order to increase throughput performance, to make it advantageous over traditional WI-Fi in ISM bands [38].

To boost the user peak data throughput, Carrier Aggregation (CA) is used in the growing broadband demanding sphere. In most studies, CA is researched in heterogeneous networks and heterogeneous radio access technology systems such as LTE and LTE-A, LTE and TVWS, LTE and WiFi, with focus on the challenges and techniques implemented to achieve aggregation [39], [12]. Aggregation is done at either the physical or MAC layers [39] and resources channel in either the downlink(DL) or uplink(UP) [12] in the 3 GPP LTE-A. Therefore, the capacity is improved with cross-carrier scheduling. Concerning alternative solutions that improve the user capacity and throughput in prominent wireless technologies, Erika et al. discussed the ability to allow coexistence of WiFi and LTE. The work uses blank sub-frames in LTE-A [5]. Alternatively, it uses guard bands to increase the bandwidth and therefore improving user experience [37].

2.6 Summary

In order to meet the objectives of this study, this section investigates work conducted in other researches in order to discover the contribution this study adds to the telecommunication domain. Previous work presents radios such as WiFi, TVWS, and WiMax to have beneficial attributes to bridging the digital divide in urban and rural areas. With low deployment costs in mind, this study adopts the use of WiFi and TVWS since these technologies utilize license-exempt bands. Although experimental measurements on WiFi and TVWS have been presented, there has been no thorough investigation made on a variety of scenarios such as buildings, vegetation, mountains affecting TVWS and no thorough comparison of WiFi and TVWS based on propagation characteristics. Experimental results presented in the literature is gathered from prediction models and simulated network environment, whereas this study is interested in the realistic effects of the environment on 5 GHz WiFi and TVWS.

CHAPTER 3

RESEARCH METHODOLOGY AND MEASUREMENT SETUPS

3.1 Introduction

The overall purpose of this work is to create and demonstrate the different performance characteristics of WiFi and TVWS using a variety of measurement metrics, following indoor and outdoor setups. Initially, the work follows the general performances of WiFi and TVWS. Secondly it addresses performance characteristics received from both these radios individually and then utilizes the characteristics to improve any "shortfalls" experienced. Research questions are posed in chapter one, and these are restated below as:

- What is the realistic outdoor link performance estimates between WiFi and TVWS?
- Can the overall network performance improve with bonding available WiFi and TVWS ?

Thus this chapter presents the equipment used; research design (indoor and outdoor setups); data collecting procedures used; data analysis tools and procedures. In this chapter, all the procedures used for the purpose of this work will be used to address the research questions formulated for this work.

3.2 Description of the Environment

The measurement process was performed at the University of Cape Town (UCT) in two locations over a few months during the summer to autumn season. The weather in Cape Town during these seasons is usually less windy and experiences less rainfall, which provides good conditions for the tests to run with little to no compromise. The first location was situated at the UCT rugby fields, where the location covered two typical rugby fields, as shown by the panoramic image in Figure 3.1. A typical rugby field covers +/- 150 m x 70 m level ground. The two fields were separated by a paved walkway. The duration of the measurement process, there was minimal distractions and no games in progress. For each setup at the rugby field, the nodes were separated at three random distances as shown by the pins on the far-right of Figure 3.2. The distances were not predetermined for the measurements at the rugby field and for the ones near the tennis field as well. The second location of interest is near the UCT tennis courts, which is located in a clump of pine trees. The positions are shown by the yellow pins on the left-most part of Figure 3.2. These locations are chosen according to the measurement scenarios that meet the LOS and NLOS requirements for this study. The selection of these locations was limited to accessible areas with open spaces and geographical features that meet a handful of scenarios specified in this study. The limitations of this study will be highlighted in the subsection below.

(a)

Figure 3.1: Panoramic view of the UCT rugby field

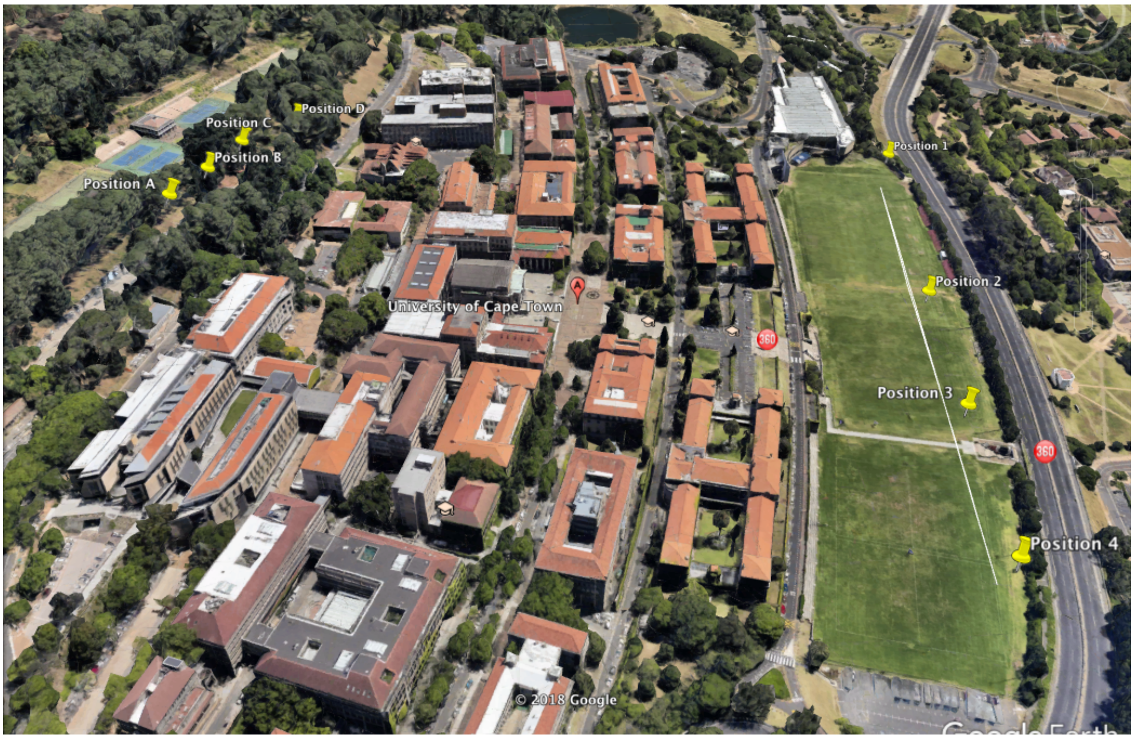


Figure 3.2: Top Google Earth view of the outdoor locations at the University' rugby fields (*right*) and near tennis courts(*left*).

(a)

(b)

(c)

(d)

Figure 3.3: Second measurement NLOS scenarios. The **X**= static node, and **Y** mobile node, with a) Presenting the first site with just one tree in between the nodes, b) A site with approximately 2-4 trees between nodes, c) A Site with approximately 4-6 trees between the nodes, and d). Site with approximately 6-8 trees between the nodes.

3.3 Types of equipment used

This work is location oriented, and measurement setups are based indoors and outdoors. The work encompasses different stages of the measurements, from when; how and why they were recorded. Each measurement setup will present characteristics that will affect the network performance, either positively or negatively. The results will be presented and analyzed in the succeeding chapter. Given that the measurements are dependent on the availability of obstructions, each setup will be clearly described, with and/or without obstacles. The main radios that are used as focal points are 5GHz WiFi and TVWS in the UHF band. In every measurement instances, there are similar devices used and they are specified as follows:

Equipment	**Quantity**	**Descriptions**
UPS power supply	2	
WSMN	2	
Laptop	2	Linux-OS
Antennas	4	WiFi and TVWS

Table 3.1: External equipment used for the setup

TVWS	**WiFi**
TVWS Radio Specifications 1. Outdoor 2. UHF supported frequency bands 3. Frequency Max : 860 MHz 4. Frequency Min : 470 MHz	WiFi Radio Specifications 1. Supports all 802.11 a,802.11 b and 802.11 g data rates 2. Type III - B miniPCi card, 6.0 cm x 4.5 cm (L x B) 3. U. FL antenna connectors on upper right corner 4. Weight : 20 g 5. Operating temperature :- 20 degrees to degrees)
Antennas **Product Details:** 1.Sealed F connector dipole housing 2. Robust 18 mm boom construction 3. Quick assembly and set up 4. "For strong signals"	Antennas **Product Details:** 1. Meraka HPN 5.1 - 5.85 GHz 2. High gain 22dBi, intergrated in enclosure 3. Manufacturer : poynting
Specifications 1. Wideband 2. Frequency R : 470-862 MHz 3. Forward gain : 9 -11.5 dB 4. Elements : 12 5. Channel numbers : 21 -69 6. Assembled length: 860 mm **WNIC:** 1. Doodle lab card DL509-78 2. TV band devices : 174 - 784 MHz	**Specifications** 1. WLM54SAG High power : 20 mW 802.11 a/b/g mini PCI card 2. 802.11 a/b/g Super AG High Power wireless mini PCI card 3. IEEE 802.11 a/b/g compatible WLAN 4. Benefits: Up to 108 Mbps (high-speed data rate) and : Up to 200 mW transmit power 5. High output power up to 23 dBm at a/b/g band 6. Dynamic frequency selection 7. 2.4 / 5 GHz IEEE 802.11 a/b/g standard **WNIC:** 1. Atheros-based 802.11 a/b/g mini PCI adadpter

Table 3.2: TVWS and WiFi electronic card specifications

(a) (b)

Figure 3.4: Doodle Lab TVWS Card

3.4 Experimental Setup

Indoor and outdoor measurements were conducted with respect to radio selections, 5 GHz WiFi and TVWS. A handful of channels were selected from specified channels that are secondarily usable, i.e. for TVWS the usable doodle lab channels in the 600 MHz band are channel 1 through 11.Doodle-lab cards uses down-converted WiFi frequencies to operate in the UHF TV band. Channel 1, 4, 7 and 11, corresponding to 540 MHz, 555 MHz, 570 MHz, 590 MHz center frequencies are chosen from the lowest, average and highest options of the usable pool of doodle lab channels for TVWS. For WiFi, the 2.4 GHz band was used and the channels selected in this band are channels 36, 40, 44 and 48 corresponding to 5180 MHz, 5200 MHz, 5220 MHz, 5240 MHz center frequencies. Each selected channel from both radios follows various parameters that will help determine the performance of each channel. The channels correspond to the center frequencies specified in Table 3.3. The performance is analyzed based on these parameters: signal strength, noise, throughput, delay, packet loss, bitrate, transmitted packets and received packets. The script depicted in APPENDIX D (Listing D.1), runs through each channel and produces data with respect to a specified parameter output from the script following the channel width and transmit power as the independent variables. Each test spans through different channels on different channel widths 20 MHz, 10 MHz, 5 MHz and transmit powers 20 dBm, 15 dBm, 10 dBm, 5 dBm.

Initial steps through this process were to conduct preliminary measurements at the Information and Communication Technologies for Development (ICT4D) lab in order to detect any errors or defects that might be caused by the devices or the setup itself. With the data collected at the ICT4D laboratory acting as a baseline for all the measurements, the data helped adjust the specified parameters to add in the script; adjust the antenna alignment and node assembly. The outdoor data revealed instability with varying channels.

Channel	Center Frequency	Case 1	Case 2	Case 3
	(MHz)	163 m	245 m	251 m
1	540	4.76	5.83	5.90
4	555	4.69	5.75	5.82
7	570	4.63	5.68	5.74
11	590	4.55	5.58	5.65
Channel	**Center Frequency**	**Case 1**	**Case 2**	**Case 3**
	(MHz)	**163 m**	**245 m**	**251 m**
36	5180	1.54	1.88	1.91
40	5200	1.53	1.88	1.90
44	5220	1.53	1.88	1.90
48	5240	1.53	1.88	1.90

Table 3.3: Fresnel radius calculations of TVWS and WiFi channels at different distances

The experiments in this work depend on Meraka White Space Mesh Nodes (WSMN) in Figure 3.7. The nodes are placed at each end of the selected geographical positions for these measurements following the location pins in Figure 3.2. Each node consists of two antennas, that is, the 5GHz WiFi and UHF TVWS antenna and the WSMN station fitted with the aforementioned boards in Table 3.2. In the set, each measurement conducted utilizes two nodes in which the nodes are labeled S and D merely for simplicity. Initially, GPS locations are recorded (using a compass app) from selected positions of interest, and weather conditions for that particular day, please refer to APPENDIX B. Node S is kept static in all setups according to an ideal location recorded, and node D is used as the mobile node, following the designated location selected depending on the measurement setup at hand.

In order to create a baseline measurement to follow while conducting the outdoor measurements, firstly the setup is assembled in the UCT computer science ICT4D laboratory. Each node is attached similar to how it is attached for the outdoor measurements as shown in Figure 3.7. Following this procedure assists in applying any necessary changes for outdoor setup. During the course of the measurement process, the outdoor setup was altered with respect to unexpected behavior following the indoor setup and thus helped improve and get the best results from this environment.

Initially while conducting the outdoor field measurements the two radios were used simultaneously, wherein WiFi is used for transmission and TVWS is then used as a "control link", and TVWS is used for transmission and WiFi as "control link". The "control link" is used for maintenance purposes, because each set of measurements features various channel widths and power levels. In each case, while one radio is used for transmission, the other radio link is used to change through the different variables (channel widths and transmit power). We encountered difficulties as the measurement process progressed due the link that was used as a "control"'s signal faded out as we tried to establish connection. Once this issue was encountered we had to use an alternative route in order to keep one radio for performance result purposes and also be able to change through the variables. Therefore, we used a Vodacom 3G modem as a trial run for the indoor setup and showed improvement to the measurement process and improved the measurement time.

We used two nodes for the duration of these measurements. For the purpose of clarity this study will follow the simple labels specified in the latter sections A and B. The indoor measurements were conducted with node S positioned on one end of the ICT4D laboratory and node D on the far end of the laboratory. Nodes S and D were positioned 21 meters (m) away from each other, and each node stretching 0.36 m below the ceiling.

Figure 3.5: The internal view of the Meraka Mesh Node

As gathered above an important section of the measurements is carried out outdo . Following what was initiated in the proposal phase of this work, there are specific scenarios of interest in order to understand the performance of these radios, i.e. clear LOS scenarios, and NLOS. With each scenario, we used 2 nodes, wherein each node consists of components highlighted under devices. The measurements were conducted at the university's rugby field, tennis court and the access control. With each location, we were able to focus the work on specific setups that helped give the work an advantageous variance. The rugby field was rarely heavily populated with people, which made a better option to capture LOS measurements. We ran three setups with varying distances between the nodes and weather conditions recorded for the purpose of noting any effects it might have on each experiment. Although weather recorded was recorded, it was not taken into account as one of the main variables for these experiments, because it is not part of the scope in these measurements. APPENDIX B gives the reader an idea of the average weather experienced for the measurement trials conducted in this location. Following this point, [31] suggest minimal attenuation (due to clouds) effect to frequencies below 10 GHz. The rugby field experiments ran over three different distances with node S kept static and node D stretched through the three different distances across the two UCT rugby fields. Each node featured a TVWS antenna that was positioned 2.34 m above ground, WiFi antenna positioned 1.84m above ground, mesh node and a 25 kg UPS (for portable power source) following the description depicted in Figure 3.7. The initial setup consisted of two nodes positioned 163 meters apart shown in Table 3.4 with the tag RG 1. Running a script from one node to another that. The variable considered with the data collected from the rugby field was only distance. The location was considerably clear and the data served as a good baseline for the rest of the experiment.

3.5 Project Design

As already described, the setup had two nodes and each node is powered by a UPS that supplies power to the node and the laptop for approximately 6 hours stretch. Each measurement experiment was conducted and maintained on a laptop (running Linux Operating System) that was connected to one node. The experiment was dependent highly on a python measurement script that follows the process highlighted in the flowchart described in Figure 3.8. The meas-

(a) (b) (c) (d) (e)

Figure 3.6: Elevation profiles for UCT Rugby Field

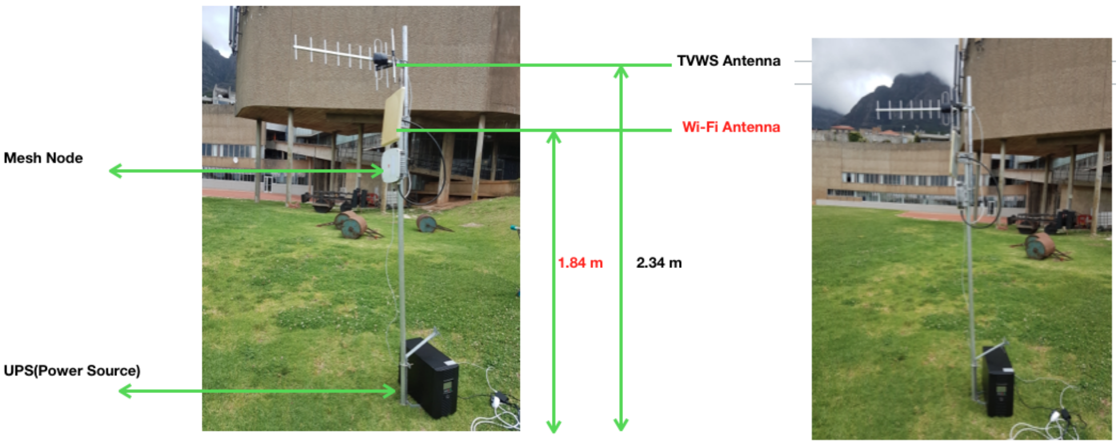


Figure 3.7: The trial setup for measuring the performance of the links at different locations with different obstructions

urement script captures throughput, signal strength, noise, transmit power, packet loss, delay, and bitrate error as dependent variables that are of interest in the work, with channel number, channel width and transmit power as independent variables. Firstly the script is run on the local node/ connected node and thus ssh into it to start the measurement experiment. The independent variables are edited into the script, i.e, "channelList", "channelWidthList",and "txPowerList" as recognized parameters in the script. The foundation of this work lies in setting up each scenario to satisfy every desired or proposed setup either indoor or outdoor measurement. The script plays a vital role in the experiment process, it helps run each measurement efficiently and store a csv file of all aforementioned parameters. The measurement script implements the step by step functions specified in Figure 3.5. Firstly we define and restructure each environmental constraints with respect to each experiment, i.e. Distance; Attenuating or Obstructing factors (Vegetation, Buildings) and Antenna polarization. We then run a script that collects data that is stored as a Comma-Separated Values (CSV) file. With each column representing information for each of the following: throughput; signal strength; packet loss; bitrate error; throughput; noise; transmit power; and delay. with respect to the dependent variables set for each row.

TAG	Location	Range	Distance	Obstructions
RG 1	Rugby Field	Position 1 - Position 2	163 m	none
RG 2	Rugby Field	Position 1 - Position 3	245 m	none
RG 3	Rugby Field	Position 1 - Position 4	251 m	none
TN 1	Near Tennis Court	Position A - Position B	30 m	1 tree
TN 2	Near Tennis Court	Position A - Position C	64 m	2 trees
TN 1	Near Tennis Court	S 33 57’ 25” / E 18 27’ 34”		2-8 trees

Table 3.4: Measurement scenarios

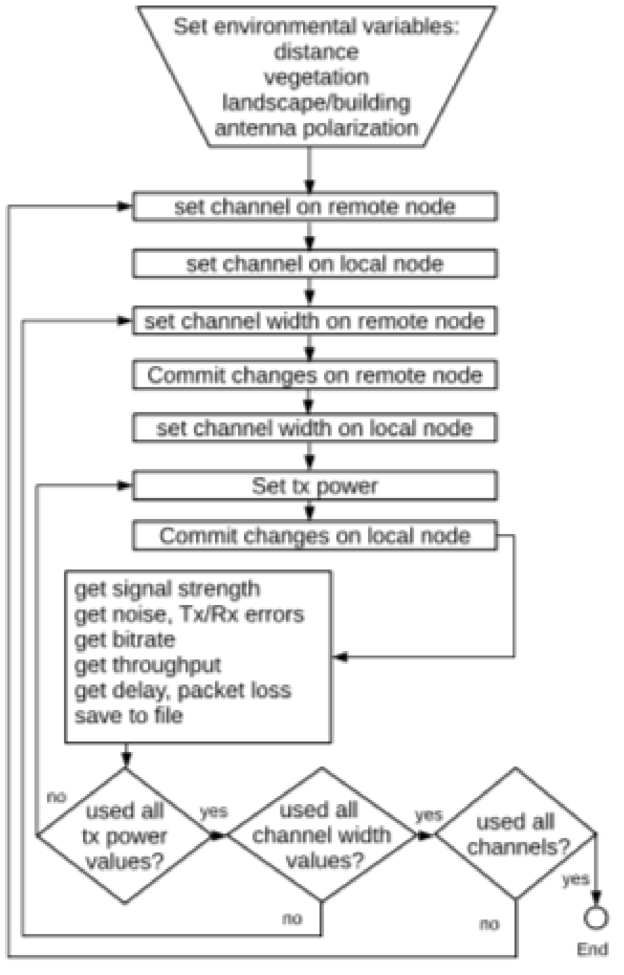


Figure 3.8: The flowchart of all the implemented steps of the script.

CHAPTER 4

SINGLE WiFi and TVWS Link RESULTS AND ANALYSIS

4.1 Introduction

The purpose of this work is to examine the behavioral performances of 2.4 GHz and 5 GHz WiFi bands and 600 MHz UHF/VHF bands TVWS radios and to use these performance traits to enhance the overall performance of the network. With this section, the overall study is pulled together to look at the problem statement and to present collected data that will assist in substantiating the hypothesis of this study. The data is collected over windy and cloudy seasons in Cape Town (between September 2017 through March 2018), and is presented and analyzed graphically below. In this study, a handful of metrics were considered to judge the link performance of each radio. They are labeled as follows (in no particular order): 1). throughput, 2). signal strength 3). packet loss, 4). transmit power, 5). noise, 6). channel width, 7). bitrate and 8). round trip time. Following the data collected in [6] to assess the quality of the link, the throughput is used as the main metric to estimate the link quality and rate selection for each radio technology. The results are collected under two scenarios: (a). transmissions in clear line-of-sight wherein the only considered interfering factor is external noise and (b) transmission setup with trees as an interfering factor. In both cases, the study considers the variance of the link quality over conditions (a) and (b). In all the trials, measurements span through 4 channels per radio, These are channels 1, 4, 7, and 11 with (540, 555, 570, and 590) MHz corresponding center frequencies for down-converted WiFi and WiFi channels 36, 40, 44, and 48 with the following corresponding center frequencies (5180, 5200, 5220, and 5240) MHz corresponding center frequencies. Throughout the experiments transmit power (TxPower) and channel width was varied per channel. These are as follows:

- Channel width = (20, 10, 5) MHz
- Transmit Power = (20, 15, 10, 5) dBm

4.2 Collected Data

The performance status of this work is presented using computed throughput averages graphically. Following some work studied under performance metrics [6], this study uses throughput as a metric for analysis. In this work, the data collected (such as signal strength, noise, bitrate etc.) along with throughput is used as a baseline information in selecting a usable channels. In addition to these metrics we use throughput as a final true measure of the performance of the link. This helps to monitor the successful transmission and reception of data packets from one point to the other over a specific communication link. Therefore the increase and decrease in this particular metric relay a very important performance behavior of the links in the network. It measures the rate of successfully delivered data packets and embeds the effect of other metrics such as Bit Error Rate (BER), signal strength and noise in the measurement.

The study presents analysis of data collected at (A). single-link LOS-Outdoor performance and (B). single-link NLOS-Outdoor performance. In order to predict the environment in which

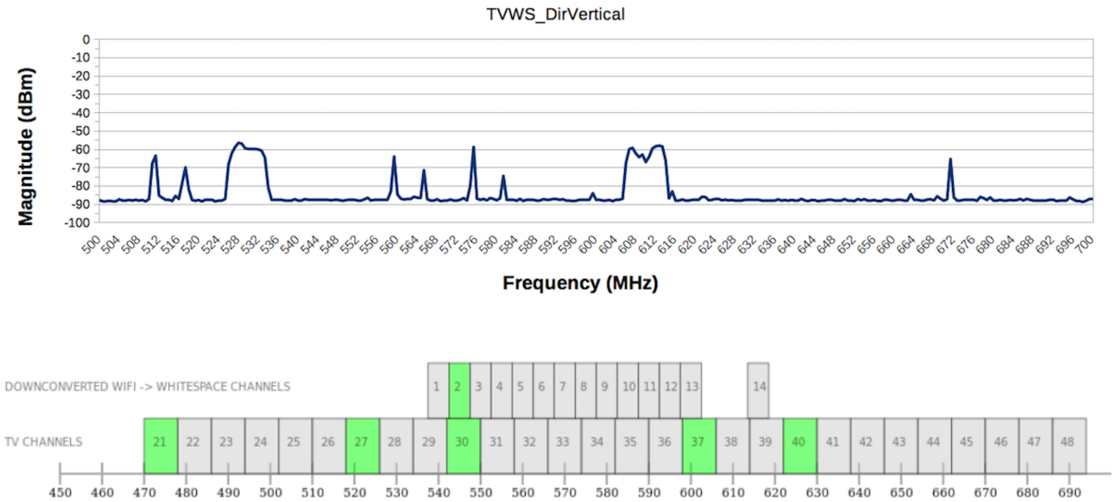


Figure 4.1: Direct vertical TVWS spectrum scans recorded from the southmost side of the rugby field.

the data was collected, the environment is initially scanned. In these particular environments, the spectrum is assesed following the locations of interests in both (A) and (B) environments. The spectrum that covers the 2.4 GHz and UHF 600 MHz frequency ranges are analyzed and presented in Figures 4.1-4.6.

(A). Single-link LOS-Outdoor Performance

Following the initial research question, this work aims to answer the question of how each radio behaves in different environmental setups, at different transmit powers, channel widths, and channels to narrow down a clear idea of how these radios differ. The work will thus try to separate the behavioral performances of WiFi and TVWS in all three locations, and then present the single link transmission of the up and downlink. There are three variables considered at different distances for these analyses, namely, *transmit power, channels, and channel width.*

The study is interested in how these radios behave. In order to gather such information, the behavior of these radios in this scenario is considered over different "*distances*". Therefore, between these two nodes, one node is kept static at one end of the environment setup and the second node is adjusted at a different position in the setup. In the spirit of simplicity, the two nodes used in this section are labeled "**S**" for the static node and "**D**" for the "mobile node". Looking at Figures 4.7 - 4.8, representing the TVWS throughput performances of channel 1 (center frequency: 540 MHz) and channel 11 (center frequency: 590 MHz), at greater measured distances, the performance gradually drops with an increase in transmit power. The inverse transmission of the data on the same link with the same power budget does not describe similar behavior. In Figure 4.7 (a) the data is transmitted from node **S** to node **D** (i.e. from position 1 - position 4 in Figure 3.2), the data portrays a gradual decrease in throughput with an increase in transmit power, and yet when data is transmitted from node **D** to node **S** (i.e. from position 4 - position 1 in Figure 3.2), there is an unexpected increase in throughput performance from 5 dBm transmit power to 10 dBm, and a sudden drop at 15 - 20 dBm power budget. This effect of transmit power over the link could be related to allowed TVWS power budget at 20 dBm. Therefore, with a high transmitted power level, the TVWS link is saturated and the performance drops. The same relationship is not translated in Figure 4.8: (a)-(b). With an average distance between the nodes,

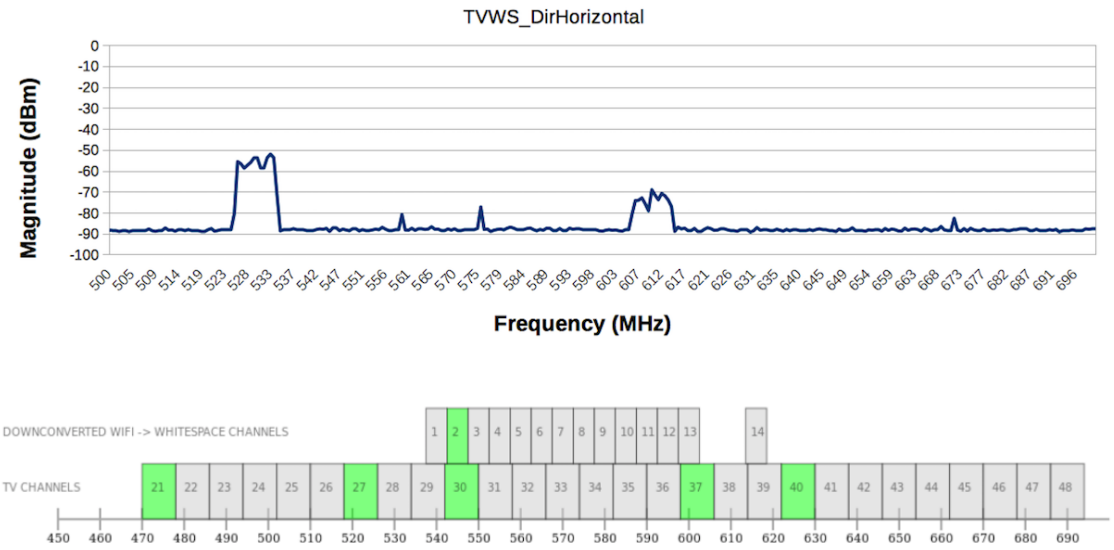


Figure 4.2: Direct horizontal TVWS spectrum scans recorded from the south-most side of the rugby field.

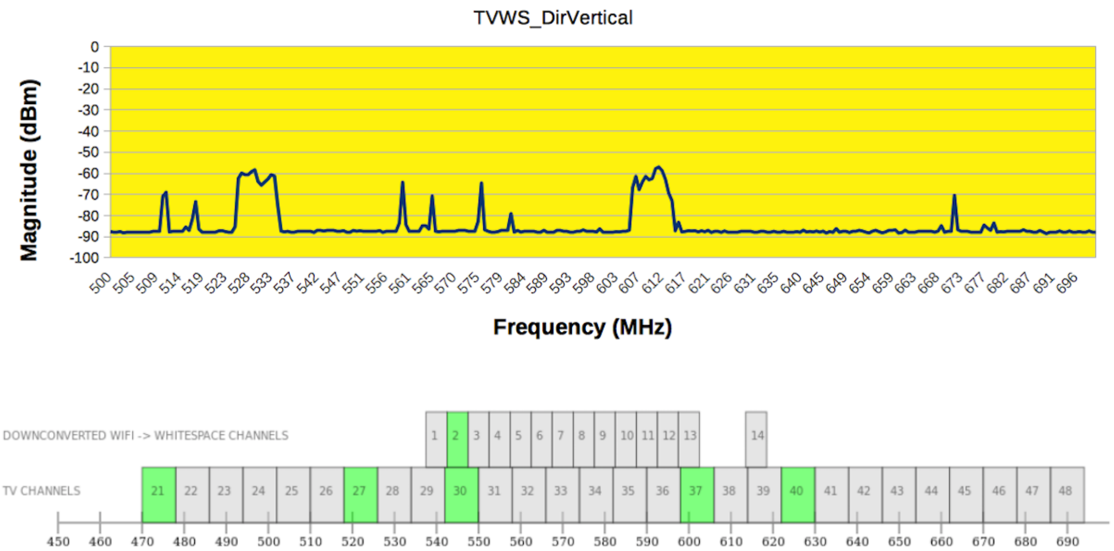


Figure 4.3: Direct vertical TVWS spectrum scans recorded from the middle of the rugby field.

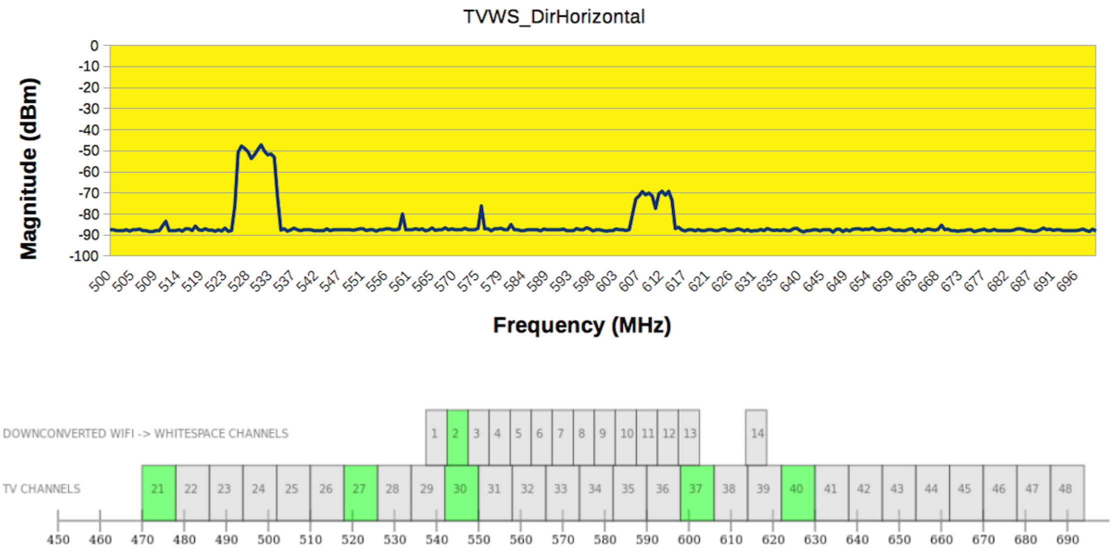


Figure 4.4: Direct horizontal TVWS spectrum scans recorded from the middle of the rugby field.

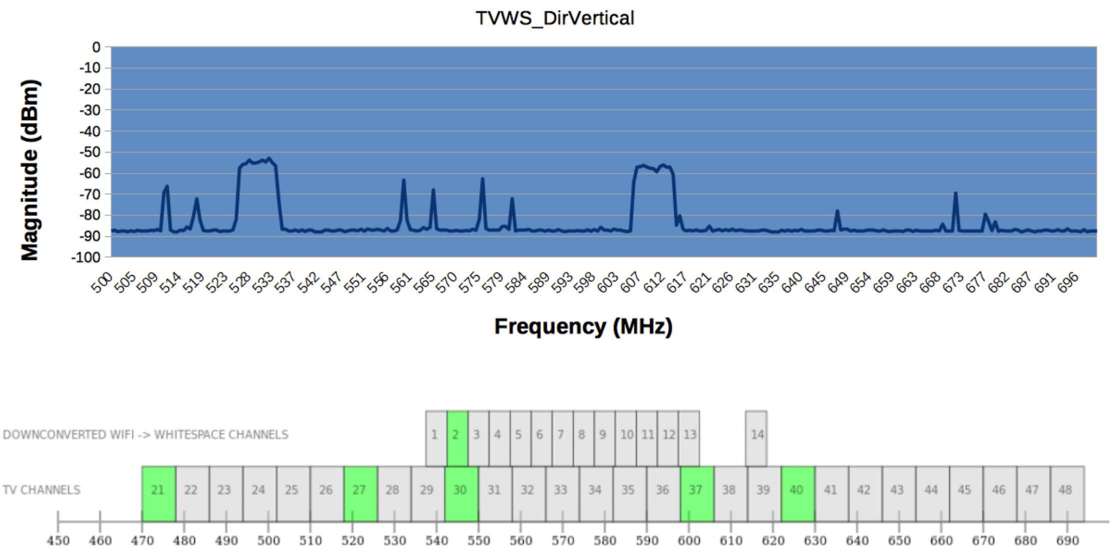


Figure 4.5: Direct vertical TVWS spectrum scans recorded from the north-most side of the rugby field.

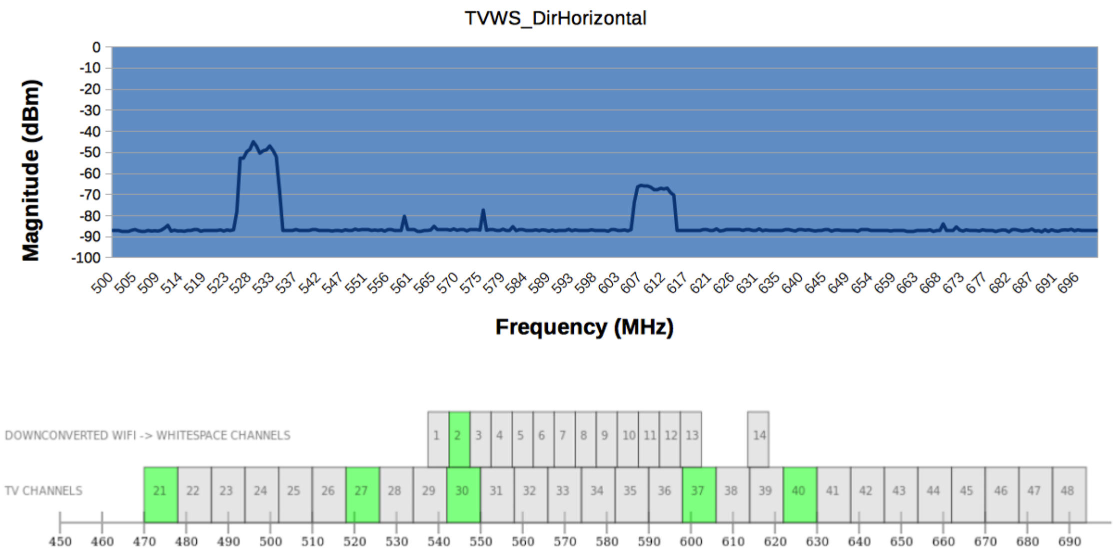


Figure 4.6: Direct horizontal TVWS spectrum scans recorded from the north-most side of the rugby field.

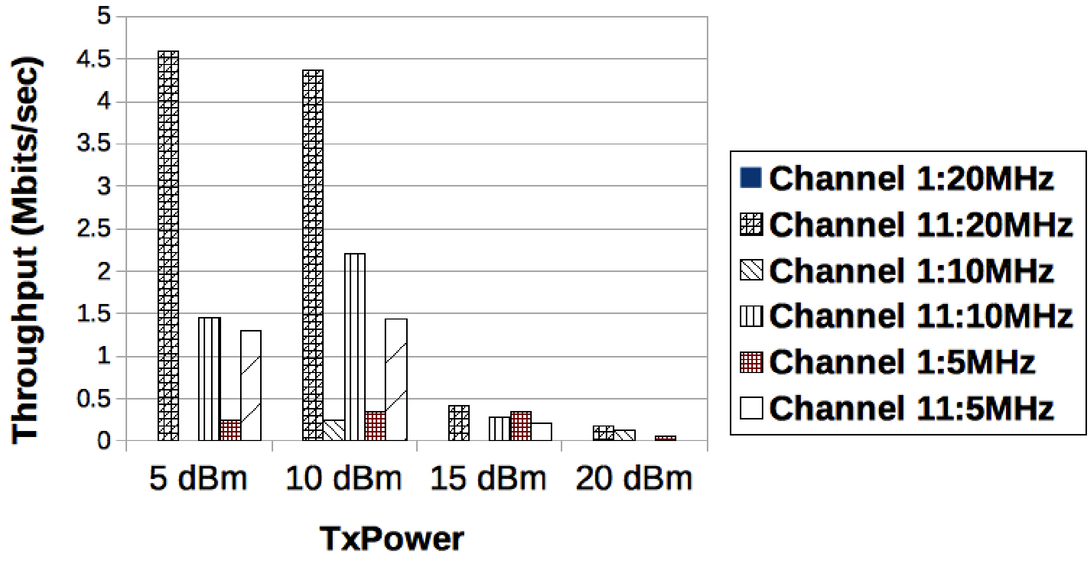
the link translates an inconsistent behavior in performance at different transmit powers at the same channel (1 and 11) set in Figure 4.7 (a and b). The link performance has a better advantage at an average transmit power in both directions, that is, from node **S** to node **D** and from node **D** to node **S**. This behavior could be related to the noise factor in the environment. As data is transmitted we notice a high noise level received by node **D**, and alternatively a lower noise level received at by node **S**. At shorter distances, channel 11 portrays a good throughput performance at 10 dBm it behaves at a standard rate in both directions. The same does not hold at greater distances, where channel 11 shows good throughput performance with 5 dBm power.

The same figures considered in the latter will be used to shed some light into how channel widths affect the link performance or quality. Channel 11 shows a better performance compared to channel 1, showing greater peaks at different channel widths and transmit powers.

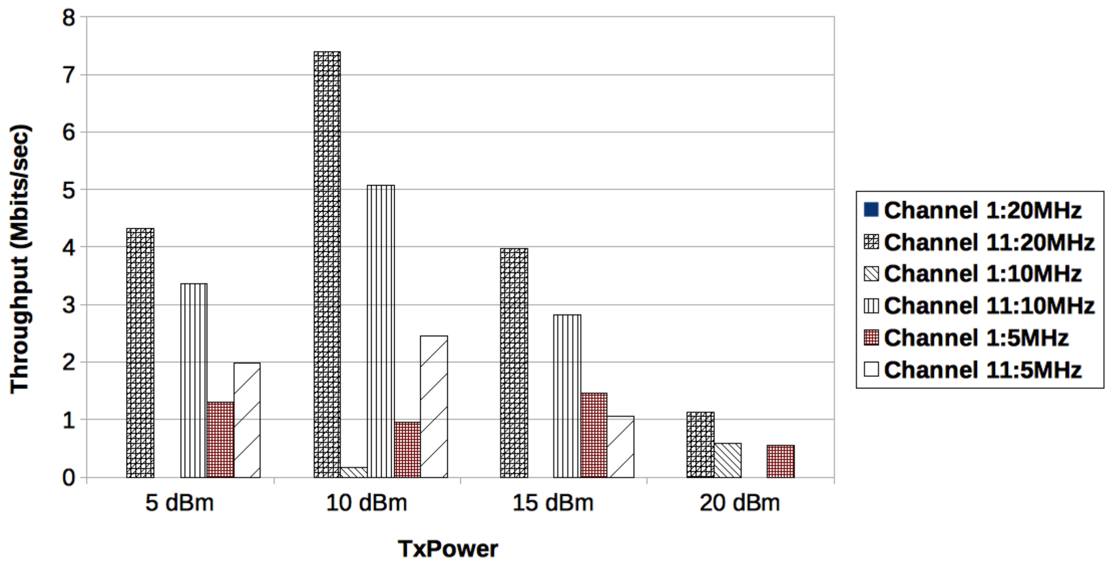
(B). Single-link NLOS-Outdoor Performance

The results presented in this sub-subsection reflect data that was collected in a grove of pine trees near the University tennis courts, depicted in Figure 3.2 on the far left with trees varying at a count of at least 1 - 8 trees in the propagation path. The data was measured through vegetation similarly to measurements collected in [13]. In this case, the results aim to present how vegetation affects the signal strength that affects the link quality in both WiFi and TVWS channels. Figures 3.3: (a)-(b) depict the environment in which the experiment was set up. The "tree count" over different distances is the only factor varied for results.

Figures 4.11-4.14 present comparisons of the network throughput at different transmit power budgets, and channel widths describing the effects of vegetation between TVWS and WiFi. In each measurement setup, the tree quantity was varied to extract the experimental data in all these radios. The set up for this experiment was not adjusted at high sites. Therefore each node's antennas were set up at similar heights depicted in Figure 3.7. The packet loss in the network is inversely proportional to the measured throughput, therefore the study presents a decrease in throughput with an increase of the packet loss in a radio. Based on Figure 4.11 and 4.14, where the throughput was measured with at least two trees between the nodes, WiFi reports a high path

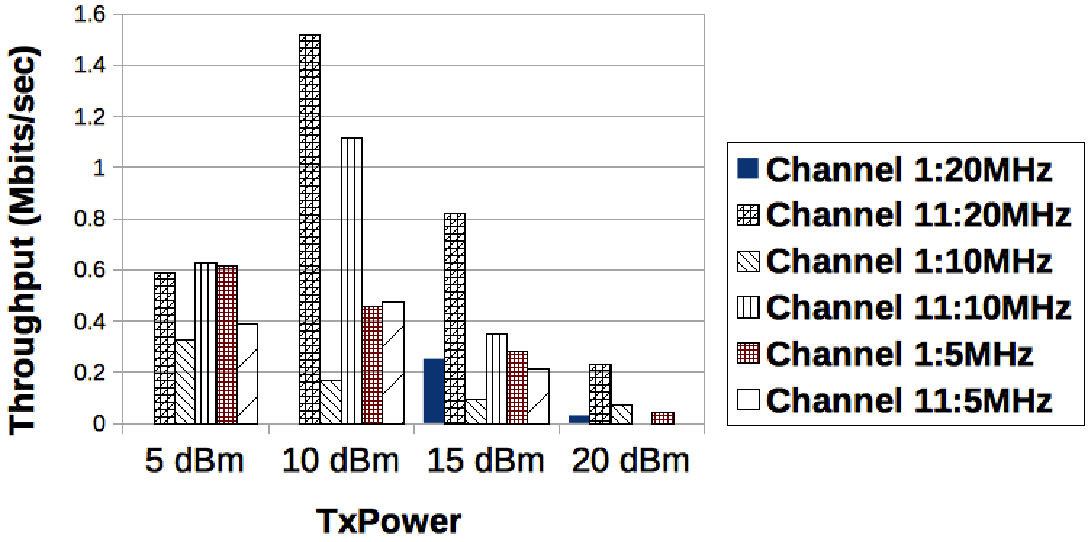


(a)

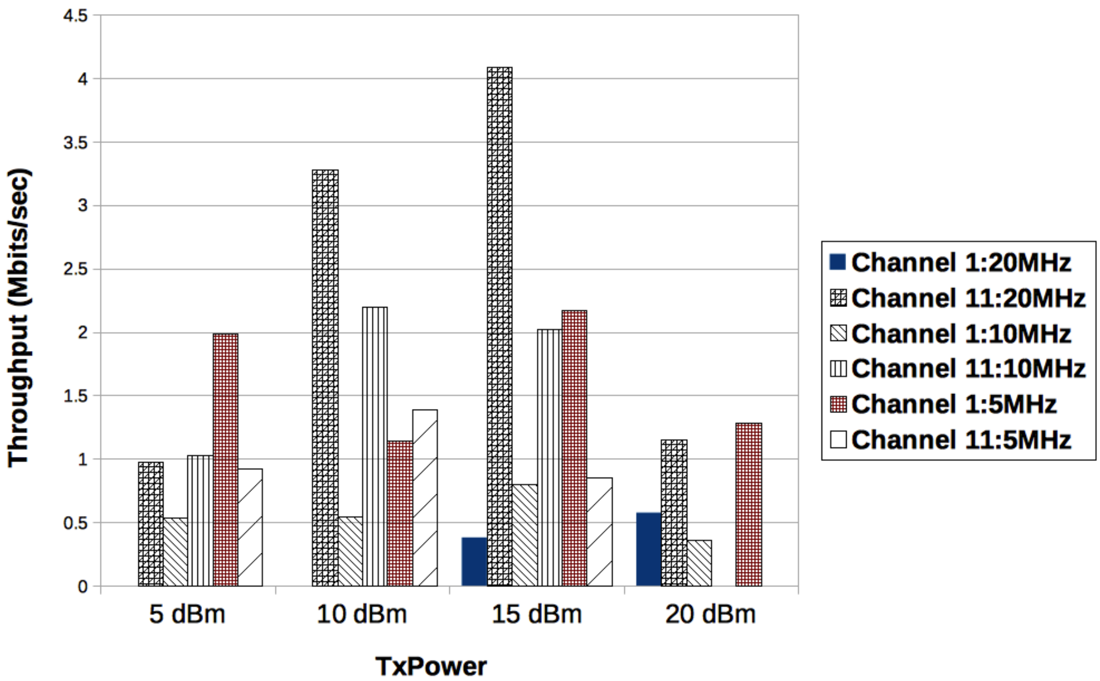


(b)

Figure 4.7: TVWS transmit power vs throughput graph recorded at the rugby field. DISTANCE: 344 m



(a)



(b)

Figure 4.8: TVWS transmit power vs throughput graph recorded at the rugby field (a). Transmission from Node S to Node D (b). Transmission from Node D to Node S. DISTANCE: 251 m

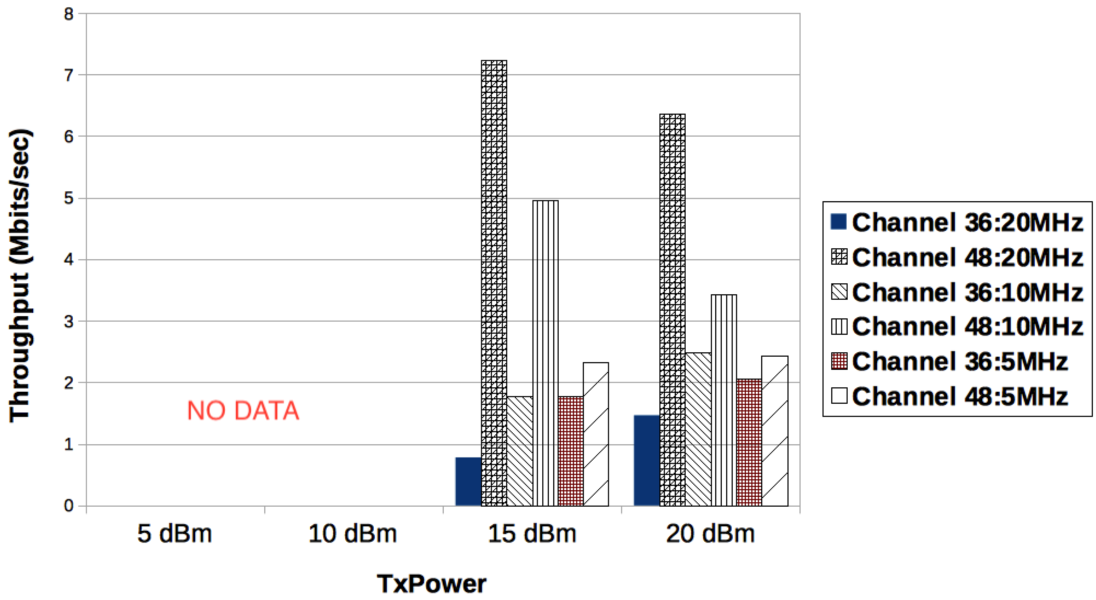


Figure 4.9: Wi-Fi transmit power vs throughput graph recorded at the rugby field. DISTANCE: 163 m

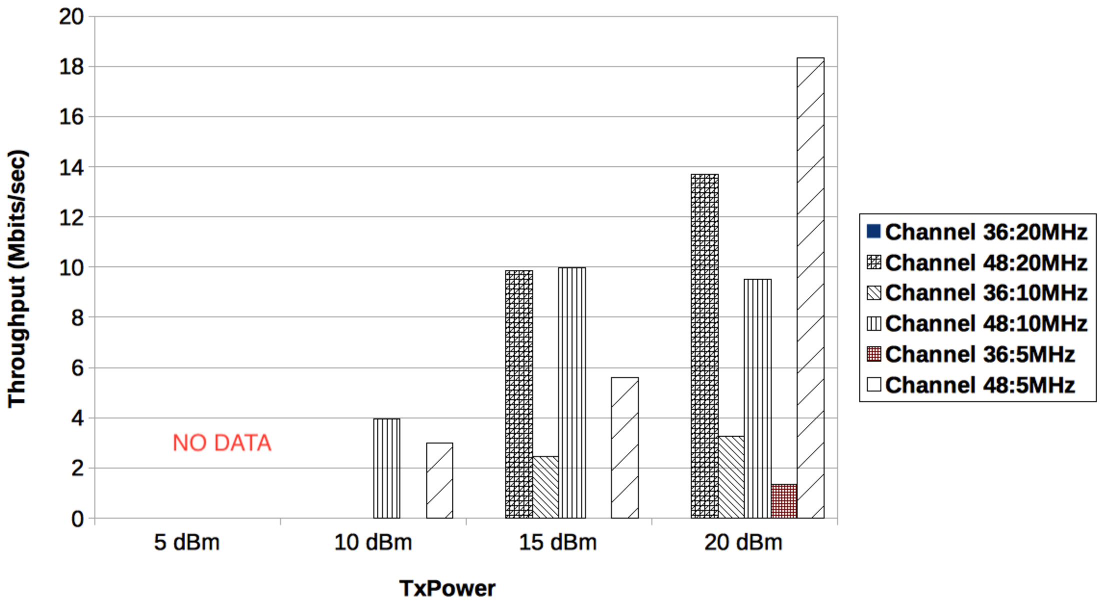


Figure 4.10: Wi-Fi transmit power vs throughput graph recorded at the rugby field. DISTANCE: 344 m

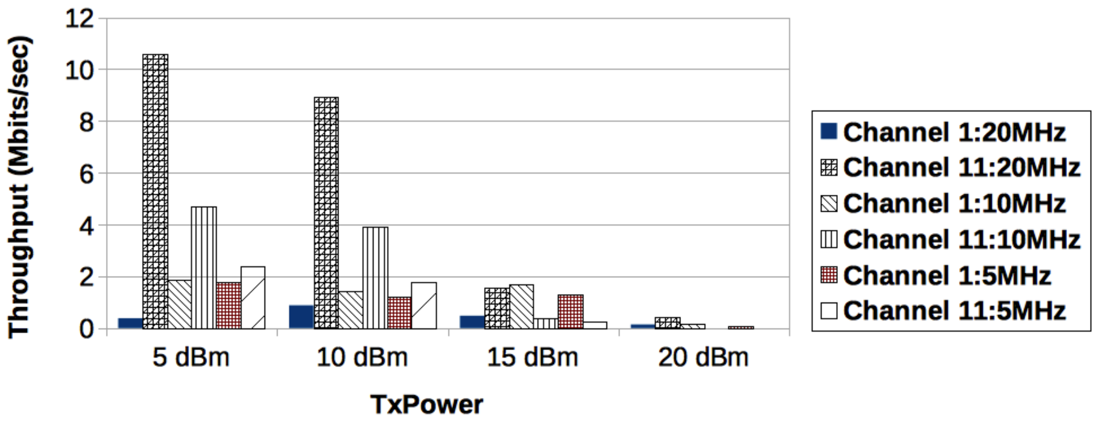


Figure 4.11: TVWS transmit power vs throughput graph recorded at the rugby field. TREE COUNT: 2

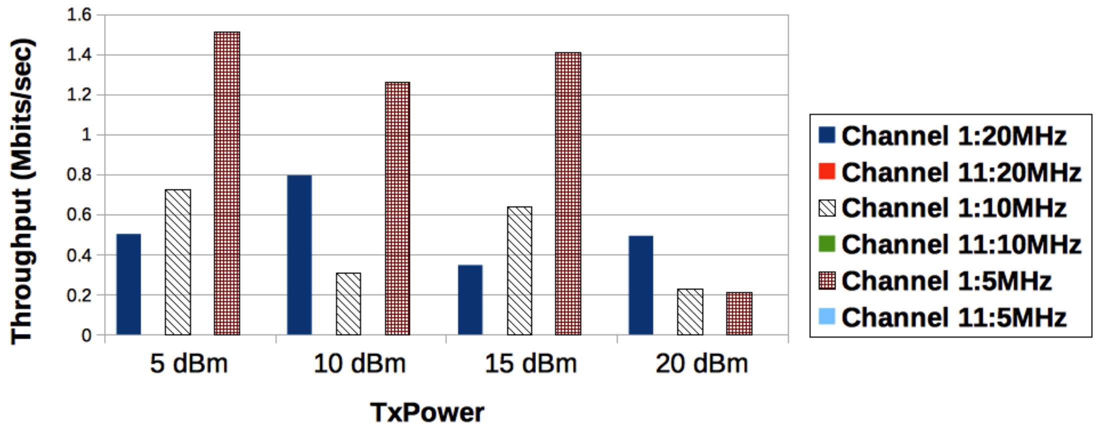


Figure 4.12: TVWS transmit power vs throughput graph recorded near the tennis court. TREE COUNT: 8

loss at lower transmit power budget, i.e. 5 dBm, compared to TVWS. WiFi presented negligible signal at setups with more than a couple of trees, in which the signal was unable to penetrate through the trees therefore the script was unable to capture any results beyond a couple of trees in this regard. Tree density poses a high attenuating effect on WiFi rather than TVWS, due to the high penetrative character of lower frequencies. Figure 4.12 presents the measured throughput in a scenario with at least 8 tree counts between the propagation path, on channel 1 and channel 11. The data present channel 1 outperforming channel 11 with high throughput output at similar set power budgets and channel width, and no throughput was recorded for channel 11 in this setup. At this maximum tree density setup compared to WiFi, a few channels presented very high path loss and fading signal, therefore yielded a bad link quality.

4.3 Discussion of the Results

In order to consider a dual operation of WiFi and TVWS in one network set up the study analyzed the behavior of both these radios in line-of-sight and non-line-of-sight (with respect to vegetation count) scenarios. In a line-of-sight scenario, the experiment adjusts the distance between the network nodes to evaluate the propagation characteristics of these radios. Adjusting the channel

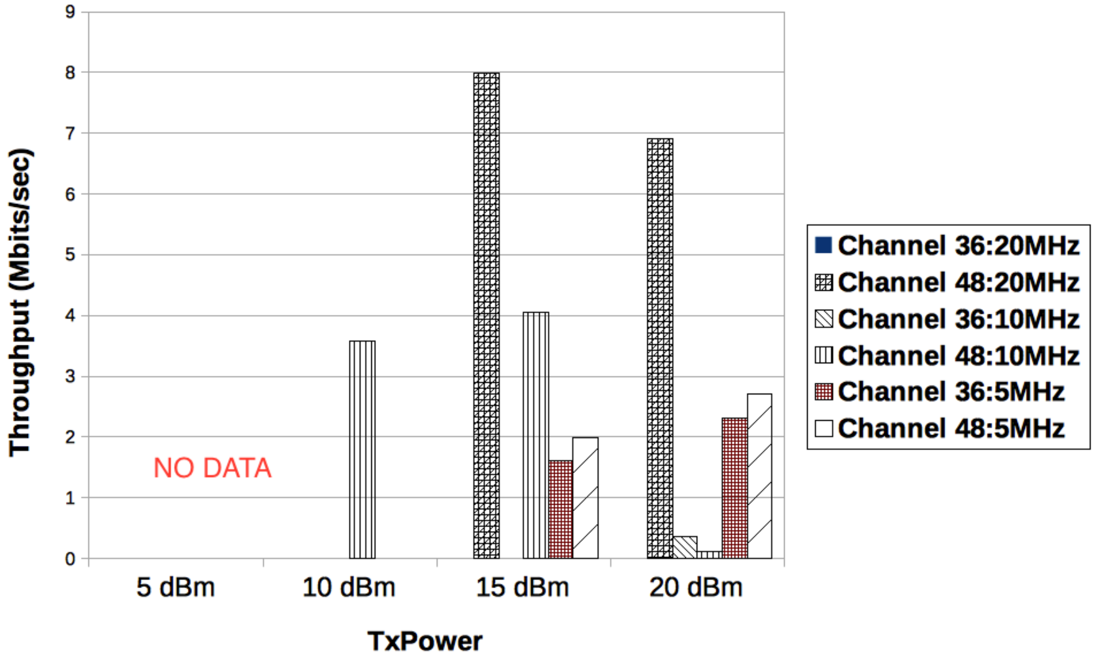


Figure 4.13: Wi-Fi transmit power vs throughput graph recorded near the tennis court. TREE COUNT: 1

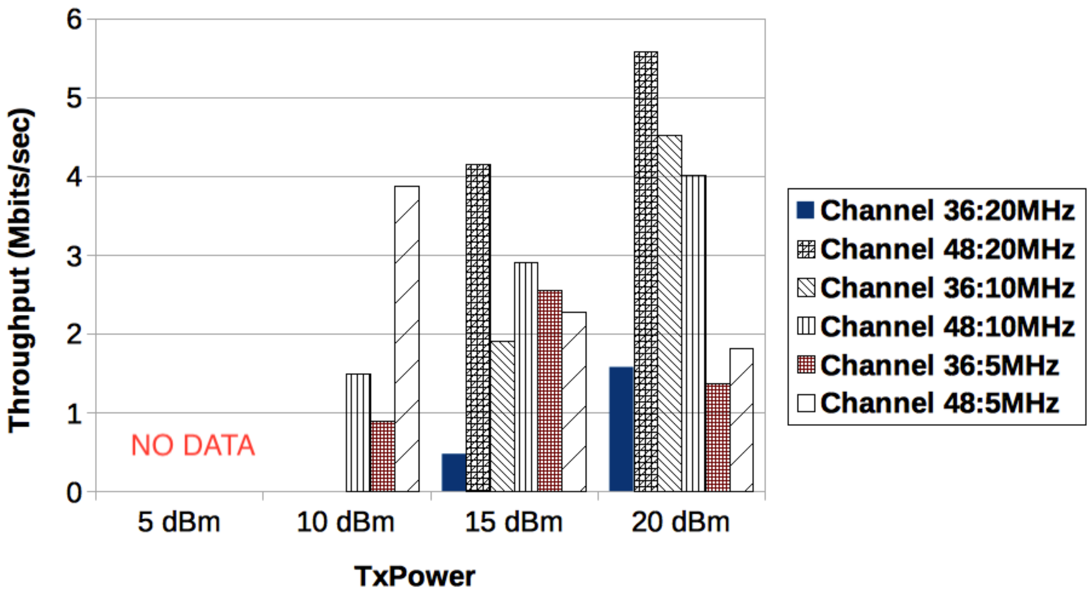


Figure 4.14: Wi-Fi transmit power vs throughput graph recorded near the tennis court. TREE COUNT: 2

width and transmit power impacts the behavior of WiFi and TVWS. Figures 4.7 - 4.14 show that WiFi performs better at high transmit powers as compared to TVWS, this could closely be related to the permissible transmit power for TVWS in South Africa and external interference in this particular area. Therefore, Figures 4.9 - 4.10 show a decrease in throughput values at 20 dBm transmit power at all distances and in order to achieve a standard throughput rate at short range distance scenarios, a 10 dBm transmit power average will be suitable. Although this is the case, the same relationship is not evident for WiFi channels, in fact, the quality of the link is degraded at lower transmit powers at short distances. Figure 4.12 shows a slight throughput increase at lower transmit power due to the increased distance between nodes in Figure 4.9, 163 meters apart and Figure 4.10, 251 meters apart. This argument can closely be related to effects this set up has on the radio propagation, this , therefore, incurs reflection, diffraction of the transmitted signal.

The purpose of running the experiments in a clear line-of-sight helps understand the relationship of the network performance in non-line-of-sight scenarios. Although WiFi shows a good performance in clear line-of-sight propagation, the link quality degrades with an introduction of external obstruction. An increase in tree count decreases signal strength in both WiFi and TVWS. However, the increment of trees between the network set up proves to affect WiFi performance over TVWS. This relates to the high penetrative quality of low-frequency bands over high-frequencies bands such as 5 GHz WiFi.

Spectrum scans show DoodleLab channels 1 and 11 corresponding to 540 MHz and 590 MHz center frequencies good for use at the UCT rugby field. In this setup channel 11 reports better TVWS performance compared to channel 1. Channel 1 reports slightly degraded links performance and this could be affected by the nearby DTV transmissions seen in Figures 4.1 - 4.6.

4.4 Summary

This chapter evaluates the propagation characteristics of WiFi and TVWS in clear line-of-sight and non-line-of-sight scenarios. This chapter considers these scenarios to achieve propagation characteristics of both WiFi and TVWS to investigate the impact network parameters have on the overall performance of the network. Channel availability plays an imperative role in allowing initial operations of WiFi and Doodle lab radio channels. Consequently, it is essential to run spectrum scans of the area of interest, with respect to free and clean channels with relatively low noise levels and adjacent channel interference. With these particular experiments from the pool of available channels to use, we worked with channels 1 and 2 for TVWS and channels 38 and 48 for WiFi.

TVWS performs better at lower transmit power and greater bandwidths. Where channel 11 proves to be a better communication link for TVWS in clear LOS. Therefore, setup in such a location would require a thorough scan of the site and with similar scan results, channel 11 would provide a better communication link with more favourable throughput with an average of at least 10 dBm. Furthermore, WiFi channel 48 provides a good throughput output over channel 36. This is evident at high bandwidths and high transmit power. Although, this might not hold in all cases, channel 48 would provide a relatively good communication link in environments with minimal noise, little to no interference and short distances.

This section aims to answer the initial research question. To understand what the realistic outdoor link performance estimates between WiFi and TVWS, and this is under conditions highlighted in this section. With the test ranges, between LOS and NLOS test setups WiFi behaves poorly in NLOS environments, where TVWS has a better advantage. This is supported by the behavior of channels 36 and 48 for WiFi over channel channels 1 and 11 for TVWS. Looking at the data presented in chapter, TVWS can be used for setups or environments with interference. Although, WiFi can shows relatively good peaks of throughput performance at high transmit powers (i.e 15 - 20 dBm) at minimal interference, it would not hold for greater distances in such environments. Therefore, TVWS provides a good communication link in such environments.

CHAPTER 5

DUAL WiFi and TVWS Link ANALYSIS

5.1 Introduction

This chapter presents measurements and analysis of single-link and aggregated links that use the 5 GHz and TVWS interfaces of the White Space Mesh Node device. To achieve aggregation of two radios, interface bonding available in the Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N)-advanced mesh protocol is used [15]. The measurements are collected in an open and short range deployment scenarios following a network setup depicted in Figure 5.1. The main objective of the experiment described in this chapter is to collect data relating to the performance of aggregated links, therefore setting up these experiment in short-range Line-of-sight scenarios benefits these experiments. The data is collected in a scenario where we can receive negligible external interference and a fair performance level for both WiFi and TVWS. With the short distance between the nodes we aim to minimize any external attenuation from obstacles and interference, hence satisfying the strength of the signals for both radios. The interference in our setup may have been limited, but not very low. The overall effect of the interference is less if the signal is strong, i.e, at shorter distances. With a short antenna pole, introducing longer distances will mean that the Fresnel zone is not clear. Thus, the short range measurements help minimize the influence of both interference and ground/obstructions. The long range measurements show how well the technology would work in practice.

5.2 Experimental Setup and Procedure

The setup for these experiments is depicted in Figure 5.1 below. WiFi channels 36, 44 and 48 corresponding to 5180 MHz, 5220 MHz and 5240 MHz center frequencies and Doodlelab card channels 1, 8, and 11 corresponding to TV white space 540 MHz, 575 MHz, and 590 MHz center frequencies are used. Transmit power was kept constant in all experiments at 20 dBm. The nodes are placed 18 meters apart with one node place on a roof 4.5 meters above ground level and the other node fixed on a ground-mounted pole, 1.7 meters above the ground. The motivation for this setup was drawn from the results in chapters 3 and 4, the experiments are arranged to ensure that the WiFi and TVWS links have sufficient signal strength to test the effect of link aggregation. The setup met the conditions of being practical (placed at a house where equipment could be kept on continuously and was secure), was sufficiently spaced to meet the far field criterion and had Fresnel zone clearance for WiFi and TVWS.

The measurements procedure is split into two parts, consisting of single link measurements and aggregated links measurements. Each test begins with setting up appropriate parameters for each experimental set. All permutations of WiFi frequencies 5180 MHz, 5220 MHz and 5240 MHz, TVWS frequencies 540 MHz, 574 MHz and 590 MHz and channel widths 5 MHz, 10 MHz, and 20 MHz are tested for each experimental iteration. For each experiment, throughput, signal

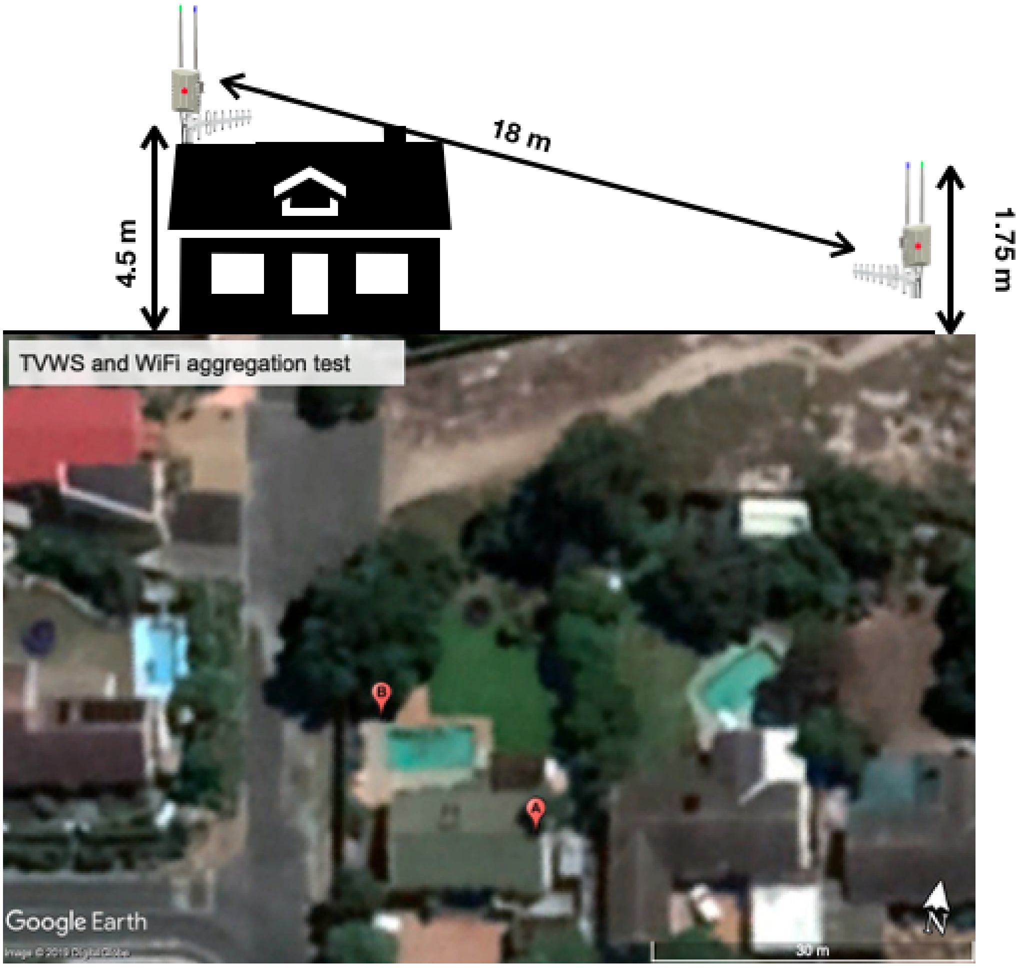


Figure 5.1: Short range aggregation network setup.

strength, noise, and round-trip time (RTT) values are collected. Measurements happen in sequence and not simultaneously. Round-trip time measurements are carried out using the ping commands that sends a ping packet every second for 10 seconds. Throughput is measured using the iperf command in TCP mode run over 0.0 - 10.0 seconds with a 43.8 KByte window size used for all setups. Signal and noise levels are measured using the iwinfo command which takes a reading every second for 10 seconds. Each experiment consisted of five trials for single and aggregated links. Statistical averages and standard deviations are calculated from these five trials.

5.3 Link Quality Analysis

Signal Strength and Noise vs Throughput

In order to evaluate the quality of single WiFi and TVWS links, the study shows an analysis of the throughput as well as signal and noise levels recorded at radio's channel. The signal quality significantly affects the performance of both WiFi and doodle lab channels in this experiment for different channels widths and channels. Therefore it is critical to analyze various permutations of channel and channels width and their impact on link performance. Table 5.1 shows signal and noise levels recorded at for 3 different channel pairs and three different channel widths of 5 MHz, 10 MHz and 20 MHz for both WiFi and TVWS. TVWS reported a high noise at different channel

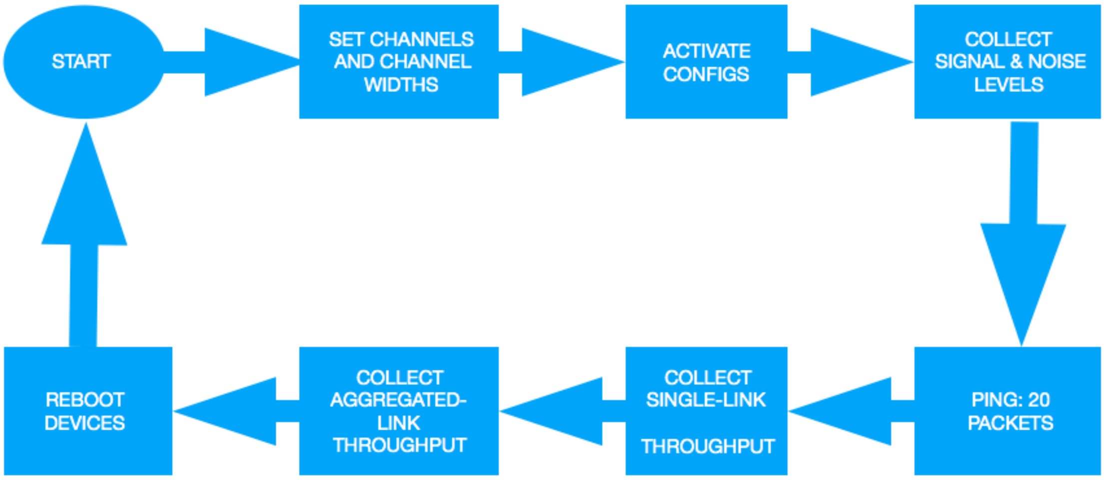


Figure 5.2: A flow of measurement procedure used in this experiment.

widths, this relationship is related to the potential external noise from other adjacent Television channels and electrical signals in the vicinity. Therefore, introducing external interference to channels as channel widths increase. The data presented in Table 5.1 is measured from the WSM node mounted on the roof of the house to the WSM node 18 meters on the ground. Therefore TVWS is more susceptible to more noise due to the adjacent channels generally used for Television broadcasting. Although the TVWS links reported more noise it does not affect the aggregated link quality.

Channel Width vs Throughput

Figures 5.3 - 5.5 below present the minimum and maximum throughput data collected from WiFi channels 36, 44, ad 48 and doodle lab channels 8, and 11. These figures 5.3 show link performance of all the channels at 5, 10 MHz and 20 MHz and all relevant permutations associated with them. WiFi shows better performance over TVWS at different channel widths. An increase in channel width shows an increase in throughput for WiFi, with TVWS showing slight improvements behind WiFi at the same power budget. There is a uniform relationship in the maximum throughput in both radios with approximately 2 Mbits/sec difference. Although, this is the case for WiFi and TVWS with a 5 MHz and 10 MHz link capacity. The figures present an increase in link capacity there is an increase in WiFi throughput performance compared to the TVWS, wherein the increase in link capacity shows a performance drawback in the TVWS. The deployment setup proves to have favorable effects on WiFi, following on the good propagation characteristics WiFi has in short range propagation. Although there is a decrease in TVWS link quality with an increase in channel width, there is a uniform difference from 5 MHz to 10 MHz capacity. At 20 MHz TVWS shows at least three times throughput difference in comparison to WFi at the same link capacity. This follows high power transmission and links capacity, the wider the capacity the more vulnerable the TVWS to interference and possibility of co-channel occupation since TVWS channels are regulated to an 8 MHz operation.

As the study analyze and present single link data, there is another significant relationship between WiFi and TVWS at different channel widths. Consider results with asymmetric channel capacity, that is, results where channel widths permutations feature different channel widths. There is a uniform throughput performance at all relative permutations in Figures 5.3 - 5.5. The results present an interesting relationship where the permutations allow an aggregation between WiFi channel width less than TVWS channel width versus when WiFi channel width being greater than TVWS channel widths. Aggregated links throughput show a better throughput performance in links where TVWS channel width is set to be lower than the WiFi channel width. In Figure 5.3 where WiFi (5 MHz) + TVWS (10 MHz) and where TVWS (10 MHz) + WiFi (5 MHz) there is an increase in throughput performance at TVWS (10 MHz) + WiFi (5 MHz). A similar relationship is presented in Figure 5.4 in similar asymmetric channel width where WiFi and TVWS occupy similar link capacities as those mentioned in the latter. TVWS shows a standard throughput performance at a 10 MHz set channel width, this holds up in all settings where it is aggregated with WiFi at either 5 MHz, 10 MHz, and 20 MHz. Figure 5.5 presents a relationship between WiFi and TVWS with respect to the use of 20 MHz link capacity. Although a wider link capacity could increase the data packets pushed into a link, this does not ultimately increase the throughput performance for all channel width selection. WiFi shows a good performance with an increase in channel width, that is, at a 20 MHz link capacity WiFi performance shows a good increase from 5 MHz and 10 MHz link capacity. To substantiate this point, Figure 5.5 shows the aggregated throughput output of WiFi and TVWS using 20 MHz capacity link. Although WiFi shows an increased throughput performance at the highest link capacity in this experiment, the link performance does not show an improvement in the link performance compared to single link performance of WiFi.

Different link capacities affect the output of throughput in a single link transmission, and therefore considering the effects of different link capacities does also affect aggregated link output. One other relationship to consider in analyzing the effects of link capacity when aggregating radios is symmetric link aggregation. The experiment considers aggregating links where channel widths are the same in both WiFi and TVWS. Figures 5.3 - 5.5 considers aggregation at 5 MHz by 5 MHz, 10 MHz by 10 MHz, 20 MHz by 20 MHz aggregated WiFi - TVWS link. Aggregated WiFi - TVWS links in Figure 5.3 show a negligible throughput improvement from a single WiFi-link, this is, there is a 0.2 Mbits increment for a single 5 MHz WiFi link throughput to a 5 MHz WiFi - TVWS ink. Doubling the link capacity does show a throughput improvement, this is, link aggregation at 20 MHz in both radios shows a +/- 10 Mbits/sec jump from aggregating the links at 10 MHz link capacity. Therefore Figures 5.3 - 5.5 shows a good link quality improvement where two 20 MHz links are aggregated from both radios.

5.4 Discussion of the Data

The main objective of this study is to investigate and lay out possible options to improve the overall performance of the network. This section's focal point is on the single and aggregate link data presented in this chapter and the previous chapter. A mechanism to make a decision can be made with respect to how environmental features, such as vegetation, affect the overall link quality of WiFi and TVWS. Decisions are also made based on aggregated link data presented in this chapter. The link quality improvement is not always guaranteed in all channel and channel width permutation, increasing the link capacity has either a positive or negative effect on the network. That is to say, there is a need for link quality considerations prior to applying link improvement approaches such as aggregation.

The main objective of link improvement is the efficient use of available resources to achieve a good or better performance than its previous link performance. Managing the performance of a link depends on the efficient use of the available resources. In order to meet these needs, we generate a threshold that will help settle a point of improvement.

In order to understand the following case studies we consider the following results between Doodlelab channel 8 and WiFi channel 36 extracted from Table E.1 in APPENDIX E as an example, with the following attributes :

Channel 36 and Channel 8:
1.Maximum aggregated throughput= 18.08 Mbits/sec

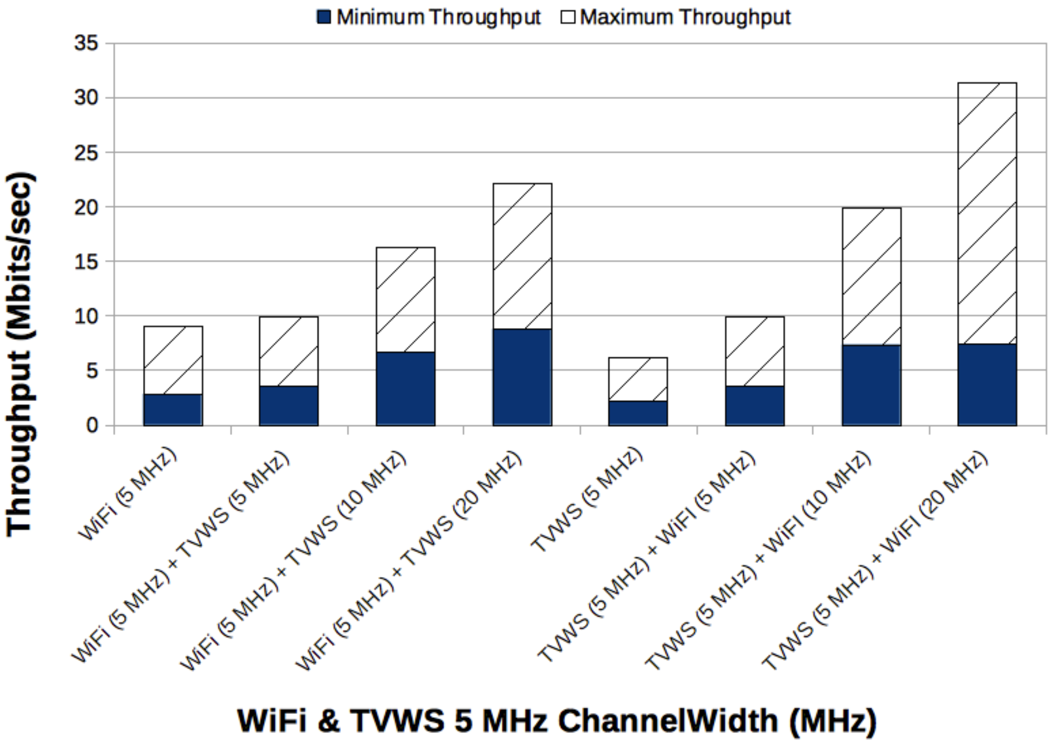


Figure 5.3: Throughput performance at a 5 MHz WiFi-TVWS link.

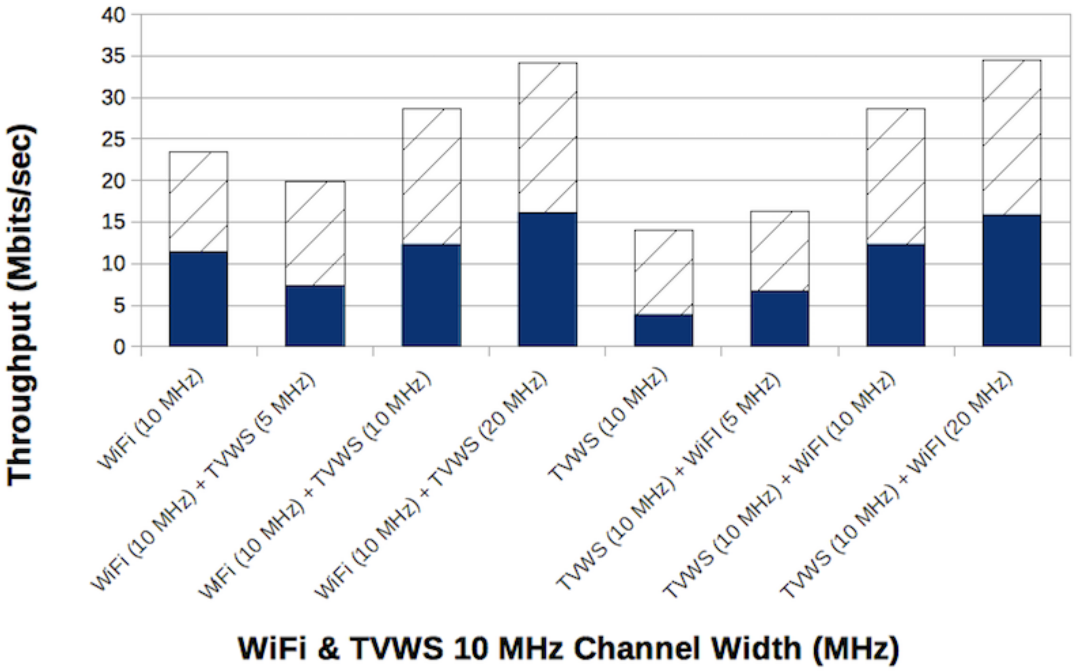


Figure 5.4: Throughput performance at a 10 MHz WiFi-TVWS link.

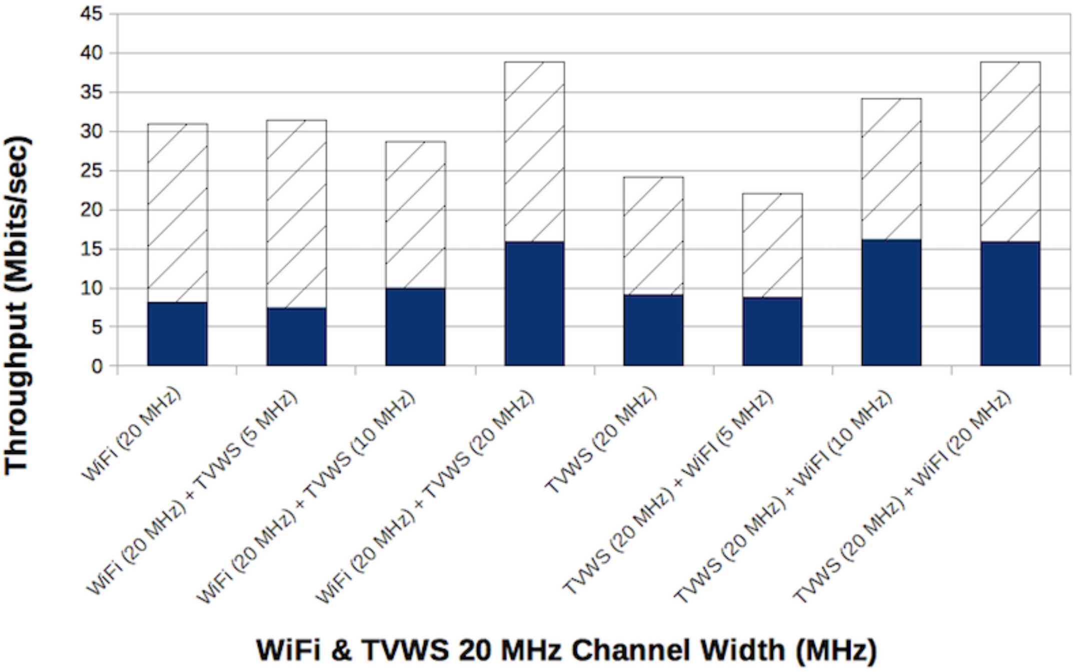


Figure 5.5: Throughput performance at a 20 MHz WiFi-TVWS link.

SingleWiFi Link (SWL) throughput= 11.58 Mbits/sec
SingleTVWS Link (STL) throughput = 14.1
With WiFi channel width set to 10 MHz and TVWS channel width set to 20 MHz.

2.Minimum aggregated throughput= 7.03 Mbits/sec
SWL throughput= 4.61 Mbits/sec
(STL) throighput= 8.58 Mbits/sec
With WiFi channel width set to 5 MHz and TVWS channel width set to 10 MHz.

Approach Number 1:

SWL= 11.58 Mbits/sec
STL= 14.1 Mbits/sec
Maximum aggregated= 18.08 Mbits/sec
Maximum aggregated/ (SWL + STL) = (18.08 / 25.68) = 70 %
Maximum aggregated/ SWL= (18.08 / 11.58) = 156 %
Maximum aggregated/ STL= (18.08 / 14.1) = 128 %

In order to consider the increase or decrease in performance, consider:
Maximum aggregated throughput/ SWL= (18.08 / 11.58) = 156% > 100% == TRUE AND Maximum aggregated throughput/ STL= 18.08 / 14.1= 128 %> 100%== TRUE,
Maximum aggregated throughput/ (SWL + STL) = (18.08 / 25.68)= 70% >= 80 == FALSE %

Approach Number 2:
SWL= 4.61 Mbits/sec
STL= 8.58 Mbits/sec
Minimum aggregated throughput= 7.03 Mbits/sec
Minimum aggregated throughput/ (SWL + STL) = (7.03 / 13.19) = 53.30 %

Minimum aggregated throughput/ SWL= (7.03 / 4.61) = 152 %
Minimum aggregated throughput/ STL= (7.03 / 8.58) = 82 %

In order to consider the increase or decrease in performance, consider:
Maximum aggregated throughput/ SWL= (18.08 / 11.58) = 152% > 100% == TRUE AND Maximum aggregated throughput/ STL= 18.08 / 14.1= 82 %> 100%== FALSE,
Maximum aggregated throughput/ (SWL + STL) = (18.08 / 25.68)= 53.30% >= 80 == FALSE %

Therefore in this example increasing the link capacity does not coherently benefit the link budget. The two approaches show two possible relationships the network portrays when aggregation is used to improve the network performance. In order to successfully utilize the radio link characteristics to benefit the overall network, we need to understand the improvement margins. The study calculates a suitable threshold to consider whether a single link transmission is better than an aggregated link transmission. Therefore a threshold level varies considerably by the relationship between an aggregated link and a single link transmission (either using WiFi and/or TVWS).

Firstly the link evaluation considered the 100 % percent single link performance margin and this will help decide on performance profits from using a single link. The aggregation threshold margin is therefore calculated at an 80 % threshold, this is due to resource management issues. As the link capacity is increased, the available resources are distributed for transmission and therefore present either low, average or high network improvement. This is to say a 50 % link improvement is a good average, but negatively impacts the available resources. Approach number 1 shows a relationship between the maximum achievable throughput in WiFi channel 36 and DoodleLab channel 8 at channel width 10 MHz WiFi channel and 20 MHz DoodleLab channel. The conclusion drawn from Approach Number 1 is that , "Maximum aggregated throughput/ SWL> 100%==TRUE AND Maximum aggregated throughput/ STL> 100%==TRUE" in this the aggregated link performance is better than the single link performance between both WiFi and TVWS.

Therefore the decision of whether to aggregate or use single links in nodes transmitting in WiFi channel 36 and DoodleLab channel 8 with 10 MHz WiFi and 20 MHz TVWS link, would be use to single links for transmissions. It is worth it to used all available resources of both WiFi and TVWS, rather than wasting the resources to aggregate. Therefore the conclusion from Approach Number 2 show a case where aggregating the links does not even play any role in improving the network, ultimately using both radios for an aggregated link budget is selfishly wasting TVWS resources.In using the single TVWS link there is a slightly better network performance.

5.5 Summary

The results show that various single link channel widths do influence the overall link quality. Investigation of symmetric and asymmetric link aggregation are presented at 20 dBm power budgets. From the analysis presented above, the aggregated link shows a better link budget where the link capacity is symmetrical. This is when channels are aggregated and 5 MHz WiFi link and 5 MHz TVWS link, 10 MHz WiFi link and 10 MHz TVWS link, and 20 MHz WiFi link and 20 MHz TVWS link. A good aggregation link capacity is seen at 10 MHz link average in both radios. Ultimately aggregating links at 10 MHz WiFi and 10 MHz TVWS, 10 MHz WiFi and 20 MHz TVWS, 20 MHz WiFi and 10 MHz TVWS, and 20 MHz WiFi and 20 MHz TVWS pass the aggregate threshold and result in good link aggregation, therefore improves the link quality without affecting the overall network and wasting resources.

CHAPTER 6

SUMMARY, GENERAL CONCLUSION

6.1 Introduction

This work was able to run full-scale performance analyses of 5 GHz WiFi and UHF 530 MHz to 600 MHz Television White Spaces working together in a network and proved that the utilization of both these radios in one network can improve the overall performance of the network. Performance characteristics are collected under two scenarios, clear line-of-sight, and near-line-of-sight. The experiments followed these two scenarios under three environment setups, i.e. indoor-LOS, outdoor-LOS, and outdoor-NLOS.With the setup and environment considered, the radios experiments span through a single link and aggregated link transmissions to determine favorable characteristics of both. Each experiment setup helped extract behavioral characteristics of WiFi and Television White Spaces (TVWS), and, therefore make informed decisions on whether to use a single link or aggregated link transmission to boost or improve the network rather.

6.2 General Conclusion

This study extensively conducted four network quality measurements under these scenarios short-range indoor line-of-sight, short-range outdoor line-of-sight, short-range range non-line-of-sight. The measurements in this study are collected using Meraka White Space Mesh Nodes that considered the dual operation of 5 GHz WiFi and UHF TVWS. The list of WLAN WiFi frequencies used is 5180 MHz, 5200 MHz, 5220 MHz, and 5240 MHz corresponding to channels 36, 40, 44, and 48. Center frequencies 540 MHz, 555 MHz, 570 MHz, and 590 MHz corresponding to DoodleLab channels 1, 4, 7, and 11. To assess the quality of link behavior the study considers network parameters such as signal strength, noise, throughput, delay, packet loss, bitrate, transmitted packets and received packets. These parameters are measured over different channels, channel widths, and transmit power. These variables satisfy the overall assessment of link transmission in all the specified scenarios. Channel is imperative for the performance given that the availability of this variable is significant for the analysis and general frequency selection and secondary channel selection. Channel width and transmit power are significant in meeting regulated transmissions settings. Throughput correlates to the bit error rate (BER), and creates a good relationship between the two. Throughput provides data that estimates the success and failures of data packet transmission and reception. The metric was used in the overall analysis of this work.

This study presents a quantitative analysis of the realistic performance estimates of WiFi and TVWS in environments that favor a line-of-sight and non-line of sight scenarios. By considering the behaviors of WiFi and TVWS in these environments, enough data is collected to generate an understanding of what to expect from WiFi and TVWS. The feedback resonates with conclusions made in the literature on WiFi performing well in short range clear line-of-sight setups and TVWS showing a rather favorable performance at longer distances and proving to have high penetrative attributes compared to WiFi, which attenuates the propagated signal between the nodes. Although

TVWS is presented in the literature as a highly favorable radio compared to WiFi, TVWS does not always produce better link qualities over WIiFi. Also, the behavior of the network does not entirely rely on one particular parameter in order to prove good link qualities. The performance of the network has to be measured and evaluated from various factors. This study does not conclude that WiFi or TVWS is more favorable than the other, instead, it can only conclude on parameters to consider in different environmental scenarios. Although TVWS is attractive for wider coverage, and better penetration in NLOS conditions it performs poorly in short-range line-of-site setups compared to WiFi.

The results report WiFi to outperform TVWS in short range line-of-sight owing to the characteristics of 5 GHz WiFi with high transmit power budgets and the result concludes on the utilization of WiFi. Although this is the case with WiFi, TVWS appears to have good link performance at 13 dBm transmit power average in short-range line-of-sight.

6.3 Limitations and Future work

The results presented in this study are only limited to the type of equipment used for these experiments. In the measurement trial, off the shelf, inexpensive cards with a limited frequency permeability range are used. Although the equipment was relevant for these particular measurement trials, the node assembly introduced some limits to measurement setup and a more wider measurement scope. Due to limited equipment for node assembly, the maximum assembled height for the set up is 2.40 meter above ground, which in return can introduce signal reflections, diffraction or even absorption depending on the frequency utilized. Therefore incur external interference to the overall link quality and received signal or even affect the Fresnel zone. The study could not explore enough environmental effects, the scope was limited to the standard pole height for travel purposes, limited travel resources, inaccurate antenna adjustment.

For future work, the author is interested in evaluating the full scope of environmental effects to 5 GHz WiFi and TVWS. Since the year 2018, TVWS operation has been regulated for use in South Africa, and thus introduce the need to extensively evaluate the full comparability of TVWS in all environments for successful network deployments and designs. Although the weather was collected prior to each measurement, the effects were not factored into the analyses. It will help create a significant evaluation of the full understanding of TVWS and WiFi, and the full extent to which the radios can operate.

6.4 Summary

To contribute to the developing countries' need on Internet access for global economic participation, the use of unlicensed wireless radio technologies are studied to foster this challenge. Thereby this study focuses on the use of license-exempt 5 GHz WiFi in the ISM bands and UHF TVWS bands to study and report on the realistic performance estimations of these technologies in these developing regions. This study presents experimental measurements that cater to scenarios typical of environmental topology. Environmental topology and scenarios vegetation, buildings and mountains having an influence on the performance WiFi and TVWS. In addition to the environment, the study also evaluated the influence of the network setup or design rather, link distance variations, node height adjustments, antenna alignments. Therefore the study concludes on 5 GHz WiFi radio to be a suitable radio technology in short range clear line-of-sight and slightly better performance in point-to-point long-range line-of-sight. The significant contribution of this work is finding a desirable point of link improvement with respect to using link aggregation or link bonding to enhance network link quality. Although 5 GHz WiFi and UHF TVWS show a very distinct propagation performance, the study suggests that complimenting the differences in these radio technology shows good link improvement and not necessarily aggregating the link budget, but rather enhancing the link quality when necessary for the network. Efficiently using the available resources to generate a threshold margin for link quality improvement.

Bibliography

[1] Khairuddin Ab-Hamid, Chong Eng Tan, and Sei Ping Lau. Self-sustainable energy efficient long range WiFi network for rural communities. *2011 IEEE GLOBECOM Workshops, GC Wkshps 2011*, (RuralComm):1050–1055, 2011. 3

[2] Mikhail Afanasyev, Tsuwei Chen, Geoffrey M. Voelker, and Alex C. Snoeren. Analysis of a mixed-use urban wifi network: when metropolitan becomes neapolitan. *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC '08*, page 85, 2008. 3

[3] Republic of South Africa. Independent Communications Authority of South Africa Notice 147 0f 2018. *Government Gazette*, 12(13):1–16, 2000. 4

[4] South Africa, The Cost, South Africa, Progressive Communications, South Africa, Broadband National Plan, and Eastern Cape. Alternatives for Affordable Communications in rural South Africa : Social Purpose GSM Licenses. 85(32939):5–8, 2010. 1

[5] Erika Almeida, Andre M. Cavalcante, Rafael C D Paiva, Fabiano S. Chaves, Fuad M. Abinader, Robson D. Vieira, Sayantan Choudhury, Esa Tuomaala, and Klaus Doppler. Enabling LTE/WiFi coexistence by LTE blank subframe allocation. In *IEEE International Conference on Communications*, pages 5083–5088, 2013. 10

[6] Michalis Faloutsos Angelos Vlavianos, Lap Kong Law, Ioannis Broustis Srikanth V. Krishnamurthy. Assessing Link Quality In IEEE 802 . 11 Wireless Networks : Which is the Right Metric ? In *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008. 21

[7] Leire Azpilicueta, Peio López Iturri, Erik Aguirre, Ignacio Mateo, and José Javier Astráin. Analysis of Radio Wave Propagation for ISM 2 . 4GHz Wireless Sensor Networks in Inhomogeneous Vegetation Environments. In *Sciforum Electronic Conference Series*, volume 1, pages 23650–23672, 2014. 8

[8] Vladimir Brik, Shravan Rayanchu, Sharad Saha, Sayandeep Sen, Vivek Shrivastava, and Suman Banerjee. A measurement study of a commercial-grade urban wifi mesh. *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC '08*, page 111, 2008. 3

[9] Irwin Brown, Boipelo Letsididi, and Mohammed Nazeer. Internet Access in South African Homes: A Preliminary Study on Factors Influencing Consumer Choice. *Electronic Journal on Information Systems in Developing Countries*, 38(2):1–13, 2009. 1

[10] Pengfei Cui, Yuanyuan Dong, Hui Liu, Dinesh Rajan, Eli Olinick, and Joseph Camp. WhiteMesh: Leveraging white spaces in wireless mesh networks. In *The 2016 International Workshop on Wireless Network Measurements and Experiments*, Tempe, AZ, USA, 2016. IEEE. 9

[11] Poulomi Das, Arindam Banerjee, and Siladitya Sen. Statistical Performance Analysis Routing Algorithms for Wimax-WiFi Integrated Heterogeneous Network. *International Journal of Modern Engineering Research*, 3(2):1032 – 1037, 2013. 2, 9

[12] Kamran Etemad, Intel Corporation, Mo-han Fong, Ravi Nory, and Robert Love. Carrier Aggregation Framework in 3GPP LTE-Advanced. *IEEE Communications Magazine*, 48(8):60–67, feb 2010. 10

[13] Mir Ghoraishi, Jun-ichi Takada, and Tetsuro Imai. Radio Wave Propagation Through Vegetation. *Wave Propagation Theories and Applications*, 2013. 25

[14] Paula Gómez-pérez and Iñigo Cuiñas. Degradation of ground-based RADAR performance due to vegetation cover. *Elsevier*, 92:230–235, 2016. 10

[15] Manuel Hachtkemper, Michael Rademacher, and Karl Jonas. Real-World Performance of current Mesh Protocols in a small-scale Dual-Radio Multi-Link Environment. *VDE ITG-Fachbericht Mobilkommunikation*, 7, 2017. 32

[16] Keith N. Hampton and Neeti Gupta. Community and social interaction in the wireless city: Wi-fi use in public and semi-public spaces. *New Media and Society*, 10(6):831–850, 2008. 3

[17] Muzaiyanah Hidayab, Abdul Halim Ali, and Khairul Bariah Abas Azmi. Wifi signal propagation at 2.4 GHz. In *2009 Asia Pacific Microwave Conference*, pages 528–531, Singapore, Singapore, 2009. IEEE. 8

[18] ICASA. The state of the ICT sector in South Africa. *Independent Communications Authority of South Africa*, (March):83, 2020. 1

[19] David Johnson, Natasha Zlobinsky, Albert Lysko, Magdeline Lamola, Richard Maliwatu, and Melissa Densmore. Head to head battle of TV White Space and WiFi for connecting developing regions. In *Africomm 2016, 6-7 Dec 2016, Ouagadougou, Burkina Faso*, 2016. 6, 9

[20] Y Kawasumi. Deployment of WiFi for rural communities in Japan and ITU's initiative for pilot projects. *Healthcom 2004, Proceedings*, pages 200–207, 2004. 3

[21] Mokwape Magdeline Lamola, David Lloyd Johnson, Albert A. Lysko, and Natasha Zlobinsky. TVWS devices Spectrum Mask Test and Analysis. *2016 Southern Africa Telecommunication Networks and Applications Conference (SATNAC 2016)*, pages 10–11, 2016. 6

[22] Richard Maliwatu, Natasha Zlobinsky, Magdeline Lamola, Augustine Takyi, David L. Johnson, and Melissa Densmore. Experimental analysis of 5 GHz WiFi and UHF-TVWS hybrid wireless mesh network back-haul links. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 261:3–14, 2019. 6

[23] Moshe T. Masonta, David Johnson, and Mjumo Mzyece. The white space opportunity in Southern Africa: Measurements with Meraka cognitive radio platform. In *AFRICOMM*, volume 92 LNICST, pages 64–73, 2012. 9

[24] C Mikeka, M Thodi, J S P Mlatho, J Pinifolo, D Kondwani, and L Momba. Malawi Television White Spaces (TVWS) Pilot Network Performance Analysis. *Journal of Wireless Networking and Communications*, 4(1):26–32, 2014. 9

[25] Chomora Mikeka, Justice Stanley Mlatho, Martin Thodi, Jonathan Pinifolo, Dereck Kondwani, Lloyd Momba, Marco Zennaro, Andrés Arcia-Moret, Carlo Fonda, and Ermanno Pietrosemoli. Preliminary performance assessment of TV white spaces technology for broadband communication in Malawi. In *Procedia Engineering*, volume 78, pages 149–154. Elsevier B.V., 2014. 9

[26] Joseph Mitola and Gerald Q. Maguire. Cognitive radio: making software radios more personal. *IEEE Personal Communications*, 6(4):13–18, 1999. 9

[27] Mohammed Nafie, Mohamed M. Khairy and Khaled Shawky. EqualizerTrainingin IEEE 802.11b Standard. *IEEE*, pages 1206–1209, 2004. 2

[28] Dusit Niyato and Ekram Hossain. Integration of WiMAX and WiFi :. *IEEE Communications Magazine*, (May):140–146, 2007. 9

[29] Frédéric Rene Rob Flickenger, Laura M. Drewet, Alberto Escudero-Pascual, Louise Berthilso, Ian Howard, Kyle Johnston, Tomas Krag, Gina Kupferman, Adam Messe, Juergen Neumann. *Wireless Networking In The Developing World.* Creative commons attribution- ShareAlike 3.0, third edition, 2013. vii, 2, 3

[30] Kajang Selangor. The study of vegetation effect on wireless streaming video for agricultural monitorig. MONITORING. *Jurnal Teknologi*, 12:69–74, 2015. 8

[31] P Series. ITU-R P.840-5 Attenuation due to clouds and fog. *Itu-R*, 5, 2012. 17

[32] S.J. Shellhammer and A.K. Sadek. Technical challenges for cognitive radio in the TV white space spectrum. *2009 Information Theory and Applications Workshop*, pages 323–333, 2009. 9

[33] Ljiljana Simic, Marina Petrova, and Petri Mahonen. Wi-Fi, but not on Steroids: Performance analysis of a Wi-Fi-like Network operating in TVWS under realistic conditions. *IEEE International Conference on Communications*, pages 1533–1538, 2012. 9

[34] STATS SA. General Household Survey 2017, Statistical Release P0318. (May):180, 2017. 1

[35] Christopher Thraves, Gilberto A. Urueta, Pablo Vidales, and Marcin Solarski. Driving the Deployment of Citywide Ubiquitous WiFi Access. *Proceedings of First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTOOLS 2008)*, 2008. 3, 8

[36] Nan Wang, Yue Gao, Yue Chen, Eliane Bodanese, and Laurie Cuthbert. Performance Evaluation of Power Control Algorithm for TV White Space Resource in UK. pages 733–736, Kun Ming,China, 2012. 2

[37] Noriyuki Yaginuma, Masahiro Umehira, Ryuhei Funada, and Hiroshi Harada. A Guard Band Aggregation Scheme for cognitive radio using TV white space. In *International Conference onCommunication Systems (ICCS)*, pages 428–432, Singapore, Singapore, 2012. IEEE. 10

[38] Yanpeng Yang, Lei Shi, and Jens Zander. On the Capacity of Wi-Fi System in TV White Space with Aggregate Interference Constraint. In *Cognitive Radio Oriented Wireless Networks (CROWNCOM), 2013 8th International*, pages 3–8, Washington, DC, USA, 2013. 10

[39] Guangxiang Yuan, Xiang Zhang, Wenbo Wang, and Yang Yang. Carrier Aggregation for LTE-Advanced Mobile Communication Systems. *IEEE Communications Magazine*, 48(2):88–93, jan 2010. 10

[40] M. Zhang and R.S. Wolff. Using Wi-Fi for cost-effective broadband wireless access in rural and remote areas. *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, (February):1347–1352, 2004. 3

[41] Zimu Zhou, Zheng Yang, Chenshu Wu, Wei Sun, and Yunhao Liu. LiFi: Line-Of-Sight identification with WiFi. *Proceedings - IEEE INFOCOM*, pages 2688–2696, 2014. 3

APPENDIX A

DOWNCONVERTED-WIFI CHANNELS CORRESPONDING TO WHITE SPACE CHANNELS

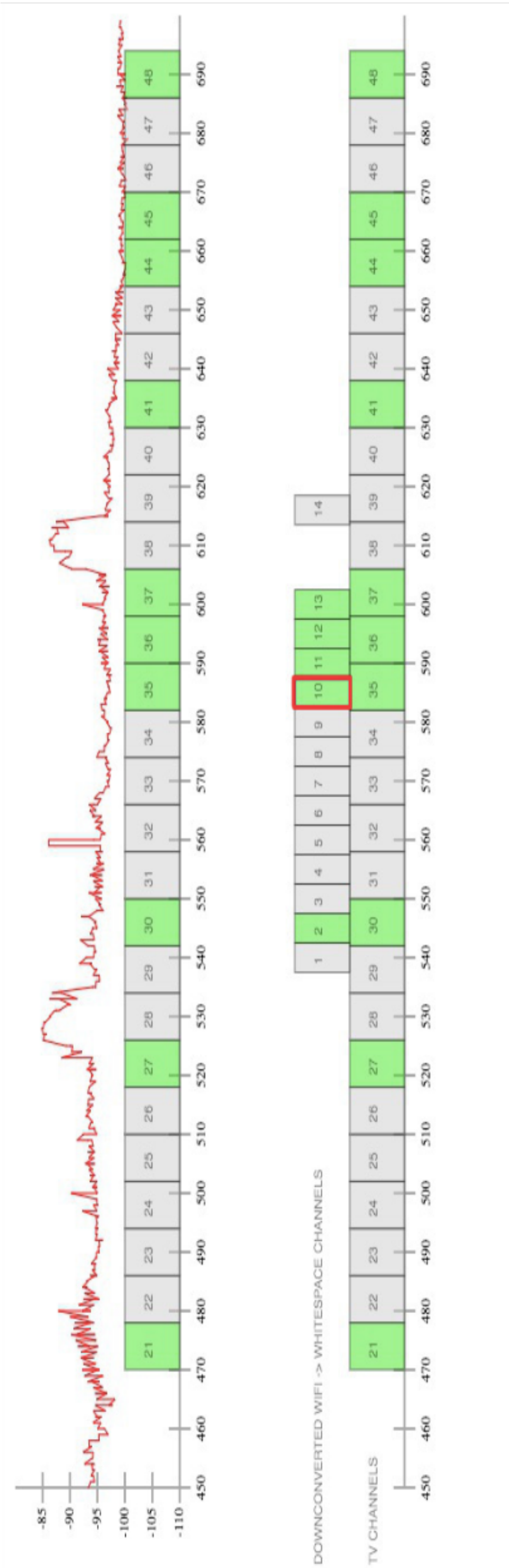


Figure A.1: Top Google Earth view of the outdoor locations at the University' rugby fields (*right*) and near tennis courts(*left*).

APPENDIX B

WEATHER DESCRIPTIONS

NB. **Node D** refers to the MOBILE NODE and **Node S** refers to the STATIC NODE

RUGBY FIELD
DAY 1: 25 October 2017(Wednesday)
WEATHER: <13-16 C|F -Humidity : 82%-Wind : 9-20 km/h >Passing Clouds
LOCATION:
Node D: S 33 57’ 27” / E 18 27’ 33”
Node S: S 33 57’ 22” / E 18 27’ 49”

DAY 2: 27 October 2017(Friday)
WEATHER: <19 C | F -precipitation : 0% -heat 48% - wind 38-49 km/h SE>
LOCATION:
Node D: S 33 57’ 27” / E 18 27’ 33"
Node S: S 33 57’ 22” / E 18 27’ 49"

DAY 3: 28 October 2017(Saturday)
WEATHER: <23 / 16 C -Wind : (19-26) km/h -Humidity : (33-36) % >Sunny
LOCATION:
Node D: S 33 57’ 27” / E 18 27’ 33"
Node S: S 33 57’ 22” / E 18 27’ 49"

DAY 4: 04 November 2017 (Saturday)
WEATHER: <19 C|F -Precip : 0% - Heat : 72 -Wind : 19 km/h > Partly Cloudy
LOCATION:
Node D:S 33 57’ 30” / E 18 27’ 47”
Node S:S 33 57’ 22” / E 18 27’ 49”

DAY 5: 06 November 2017 (Monday)
WEATHER: <21 C|F -Precip : 0% -Heat : 59 -Wind : 45 km/h >Cloudy
LOCATION:
Node D:S 33 57’ 30” / E 18 27’ 47"
Node S:S 33 57’ 22” / E 18 27’ 49"

DAY 6: 08 November 2017 (Wednesday)
WEATHER: <27 C|F-Precip : 0% -Heat : 40 -Wind : 32 km/h >Cloudy
LOCATION:
Node D:S 33 57’ 33” / E 18 27’ 42”
Node S:S 33 57’ 22” / E 18 27’ 49”

TENNIS COURT
DAY 7: 16 November 2017 (Wednesday)
WEATHER: <20 C -Precip : 0% -Humidity :40% -Heat : 61% -Wind : 38-49 km/h >Partly Sunny
LOCATION:
Node D:S 33 57’ 26” / E 18 27’ 32”
Node S:S 33 57’ 25” / E 18 27’ 32”

DAY 8: 17 November 2017 (Wednesday)
WEATHER: <23 C-Precip : 0% -Humidity : 35% -Wind : 12-28 km/h >Sunny
LOCATION:
Node D:S 33 57’ 26” / E 18 27’ 32"
Node S:S 33 57’ 25” / E 18 27’ 32"

DAY 9: 23 November 2017 (Thursday)
WEATHER: <21 C-Precip : 0% -Heat : 62% -Wind : 24 km/h >Sunny
LOCATION:
Node D:S 33 57’ 26” / E 18 27’ 32"
Node S:S 33 57’ 25” / E 18 27’ 32"

DAY 8: 24 November 2017 (Friday)
WEATHER: <21-22 C-Precip : 10% (dropped to 0%) - Heat : 64% -Wind : 40 km/h(dropped to 27 km/h) >Cloudy
LOCATION:
Node D:S 33 57’ 26” / E 18 27’ 3"
Node S:S 33 57’ 25” / E 18 27’ 32"

DAY 9: 27 November 2017 (Monday)
WEATHER: <17 C|F-Precip : 4% - Heat : 83% -Wind : 34 km/h >Mostly Sunny
LOCATION:
Node D:S 33 57’ 26” / E 18 27’ 3"
Node S:S 33 57’ 25” / E 18 27’ 32"

DAY 10: 28 November 2017 (Tuesday)
WEATHER: <21 C|F-Precip : 0% - Heat : 65% -Wind : 43 km/h >Cloudy
LOCATION:
Node D:S 3357’25” / E18 27’ 32”
Node S: S 33 57’ 24”/ E 18 27’ 34”

DAY 11: 29 November 2017 (Wednesday)
WEATHER: <25 C|F-Precip : 0% - Heat : 56% -Wind : 42 km/h >Cloudy
LOCATION:
Node D:S 3357’25” / E18 27’ 32”
Node S: S 33 57’ 24”/ E 18 27’ 34”

DAY 12: 19 February 2018 (Monday)
WEATHER: <18-25 C A few Clouds | Wind Speed: 16 km/h)
LOCATION:
Node D:S 3357’25”/ E18 27’ 32”
Node S:S 33 57’ 24”/ E 18 27’ 34"

APPENDIX C

LINK BUDGET CALCULATIONS FOR WiFi and TVWS

Please note:

$\mathbf{P}_{tx}$: Transmit Power in dBm

$\mathbf{G}_{tx}$: Antenna Gain in dBi

EIRP : Equivalent Isotropically Radiated Power

$\mathbf{P}_{inc}$: Incident Power density in dBm

$\mathbf{P}_{rx}$: Receiver Power

LINK BUDGET TABLE

VARIABLES	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS	TVWS
Ptx (dBm)	20	20	20	20	20	20	15	15	15	15	15	15	10	10	10	10	10	10	5	5	5	5	5	5
Gtx (dBi)	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
EIRP (dBm)	28	28	28	28	28	28	23	23	23	23	23	23	18	18	18	18	18	18	13	13	13	13	13	13
Frequency (MHz)	540	590	540	590	540	590	540	590	540	590	540	590	540	590	540	590	540	590	540	590	540	590	540	590
Distance (m)	163	163	245	245	251	251	163	163	245	245	251	251	163	163	245	245	251	251	163	163	245	245	251	251
Loss FSL (m)	71.3	72.1	74.9	75.6	75.1	75.9	71.3	72.1	74.9	75.6	75.1	75.9	71.3	72.1	74.9	75.6	75.1	75.9	71.3	72.1	74.9	75.6	75.1	75.9
Pinc (dBm)	-43.3	-44.1	-46.9	-47.6	-47.1	-47.9	-48.3	-49.1	-51.9	-52.6	-52.1	-52.9	-53.3	-54.1	-56.9	-57.6	-57.1	-57.9	-58.3	-59.1	-61.9	-62.6	-62.1	-62.9
Prx (dBm)	-35.3	-36.1	-38.9	-39.6	-39.1	-39.9	-40.3	-41.1	-43.9	-44.6	-44.1	-44.9	-45.3	-46.1	-48.9	-49.6	-49.1	-49.9	-50.3	-51.1	-53.9	-54.6	-54.1	-54.9
	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi	WiFi
Ptx (dBm)	20	20	20	20	20	20	15	15	15	15	15	15	10	10	10	10	10	10	5	5	5	5	5	5
Gtx (dBi)	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
EIRP (dBm)	43	43	43	43	43	43	38	38	38	38	38	38	33	33	33	33	33	33	28	28	28	28	28	28
Frequency (MHz)	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240	5180	5240
Distance (m)	163	163	245	245	251	251	163	163	245	245	251	251	163	163	245	245	251	251	163	163	245	245	251	251
Loss FSL (m)	91.0	91.1	94.5	94.6	94.7	94.8	91.0	91.1	94.5	94.6	94.7	94.8	91.0	91.1	94.5	94.6	94.7	94.8	91.0	91.1	94.5	94.6	94.7	94.8
Pinc (dBm)	-48.0	-48.1	-51.5	-51.6	-51.7	-51.8	-53.0	-53.1	-56.5	-56.6	-56.7	-56.8	-58.0	-58.1	-61.5	-61.6	-61.7	-61.8	-63.0	-63.1	-66.5	-66.6	-66.7	-66.8
Prx (dBm)	-25.0	-25.1	-28.5	-28.6	-28.7	-28.8	-30.0	-30.1	-33.5	-33.6	-33.7	-33.8	-35.0	-35.1	-38.5	-38.6	-38.7	-38.8	-40.0	-40.1	-43.5	-43.6	-43.7	-43.8

APPENDIX D

MEASUREMENT SCRIPTS USED FOR TESTING SINGLE AND BONDED LINKS

```
# Authors: Natasha Zlobinsky, Richard Maliwatu, David Johnson
# version: 0.7
# First released: 26 January, 2016
# Last revised: 24 March, 2019
#
# DESCRIPTION: this script measures the throughput, signal strength, and packet
    error
# for a given combination of channel, channel width, and tx-power values
#
# TROUBLESHOOTING: The script uses ssh to set parameters on the remote host.
# So  before running script ensure that there is connectivity between the two nodes
    ,
# check that the channel and channel width are the same on both ends,
# start iperf server manually: iperf -s
# we refer to the host running iperf server as the remote host.

# Time required to run the script to completion is approximately  67 minutes
# for four channels, three channelwidth values and four txpower values, and iperf
    timeout set to 150 seconds, ping packet count of 20
#
#
# REVISION NOTES:
# 0.6:
# added provision to use 3g modems to establish a control link
# 0.5:
# instead of geting signal strength and noise at a given instance, we get multiple
    samples using a bash script while iperf is running
#
# 0.4:
# (i) we use iperf with the -r option to get throughput in both directions
# (ii) we also get the signal strength and noise for both local and remote node?
# (iv) To deal with the problem of measurement process breaking because a link
    broke, when measuring performance of
#      WiFi, we use the TVWS link to connect to the node and changes WiFi link
    settings, and vice versa when measuring
#      performance of TVWS. This idea can easily be extended to use as an example,
    LTE link for control purposes.
#      In previous versions, we used a single link for everything, which proved
    problematic whenever
#      the link broke.
# (v) For signal strength and noise, read <what?> file directly instead of reading
    iwconfig output?
#
# 0.3
# we added ping command to get packet loss and round trip time (RTT)
#
```

```
# 0.2:
# Instead of sleeping in between commnds, this version runs command on remote host and then run command on local host
# Then go back to remote host. The idea is that the time between switching from remote node to local host and back to remote host
# is adequate for the script to run without triggering kernel panic. Furthermore, this version uses pexpect to include ssh password in the script
#
#
# INSTRUCTIONS:
# 1) Make sure there is connectivity between the nodes
# 2) Edit lines 128-130 for parameter combinations of interest.
#    For the script to run to completion smoothly, keep channelList at two channels atmost
# 3) pull up terminal on laptop and ssh into "local" node
# 4) from "local" node terminal, ssh into "remote" node
# 5) Pull up another terminal instance on laptop and ssh into local node.
#    *At this stage you have two terminals from which to control the two nodes from.
# 6) start iperf server on remote host as follows:
#               (i) iperf -s -f m
#              (ii) CTRL + Z

#              (iii) bg
#              (iv) exit
#             To terminate the process use: kiall -s kill iperf
#
# step (ii) pauses iperf server; step (iii) puts the paused process in the background and resumes it.
# Step 6 (i-iv) is aimed at keeping the iperf server running even after ssh session is terminated.
# Alternatively, iperf can be launced to run in the background as follows: iperf -s -D
# 7) Run the script on the local host as follows:

#  python  tool_v5.py -i <interface(local)> -s <iperf server ip> -c <controlLink> -r <Interface(remote)> -p <ssh password> -o <outfile>

# OUR CASE SPECIFIC NOTES:
# 1) Run script from node labelled "Ocean View" or "masi" because it already has python 2.7 installed. Then use other node as "remote" node.
#
#
# TODO LIST
# 1) save output to file on each iteration in case script terminates ubruptly
# 2) handle case when signal samples file contains "unknown dBm" text
#
# 2) consider using iperf for packet loss, delay stats? (this is currently only feasible when iperf is run in "udp throughput" mode)
# 3)
# 4) try '-d'  iperf options for simultaneous bi-directional throughput. NOTE: This requires a multithreaded version of iperf
# 5) include udp throughput?

import pexpect
import subprocess
import sys
import re
import csv
import getopt
import time
import signal
import math

#get the runtime arguments, display usage instructions
#in future use argparse module for elegant command line argument handling instead of sys.argv
```

```

def usage():
    print 'USAGE SYNTax: '+ sys.argv[0]+' -i <interface>  -s <iperfServer IP> -c <controlLink> -r <remoteInterface> -p <sshPassword> -o <outfile>'

argv = sys.argv[1:]
if len(sys.argv) < 12: #spaces between options are counted too
    print"MISSING SOME ARGUMENTS!"
    usage()
    sys.exit(2)
else:
    print "TOO MANY ARGUMENTS SUPPLIED!"
    usage()
    sys.exit(2)

interface = '' #wireless interface to run  measurments on
iperfServer ='' #iperf server IP address
remoteInterface = ''#wireless interface on remote host
filename = '' #file to save to
controlLink = '' #ip address of remote node to use for control purposes

try:
    options, args = getopt.getopt(argv, "h:i:s:c:r:p:o:", ["help","interface=", "iperfServer=", "controlLink=", "remoteInterface=", "sshPassword=", "outfile="])
except getopt.GetoptError:
    print "INVALID ARGUMENTS SUPPLIED!"
    usage()
    sys.exit(2)
for opt, arg in options:
    if opt in ("-h", "--help"):
        usage()
        sys.exit(2)
    elif opt in ("-i", "--interface"):
        interface = arg
    elif opt in ("-s", "--iperfServer"):
        iperfServer = arg
    elif opt in ("-c", "--controlLink"):
        controlLink = arg
    elif opt in ("-r", "--remoteInterface"):
        remoteInterface = arg
    elif opt in ("-p", "--sshPassword"):
        sshPassword = arg
    elif opt in ("-o", "--outfile"):
        filename = arg
    else:
        print "INVALID USAGE!"
        usage()
        sys.exit(2)


# The interface indices listed below are specific to our current radio configuration
# Useful when setting things using uci command  e.g. uci set wireless.@wifi-device[2].channel=40
if interface == "wlan1-mesh": # WiFi 5GHz Panel
    ifINDEX = "1"
elif interface == "wlan2-mesh":
    ifINDEX = "2"
elif interface == "wlan3-mesh": # TVWS
    ifINDEX = "3"
if remoteInterface == "wlan1-mesh":
    remote_ifINDEX = "1"
if remoteInterface == "wlan2-mesh":
    remote_ifINDEX = "2"
elif remoteInterface == "wlan3-mesh":
    remote_ifINDEX = "3"

```

```
#ipinterface = interface + "_13"
ipinterface = "bat0"

remoteipInterface = remoteInterface + "_13"


#list of channel, channel widths, and txpower values to work with
#channelList = ['36', '40', '44', '48'] #use values as strings "" for easy concatenation with cmd

#**************************************************************************
# Pre-populate WiFi and TVWS channels to work with
# The assumption is that the nodes are configured identically
# i.e. wlan2 and wlan3 are the WiFi and TVWS interfaces respectively on both nodes
#****************************************************************************
if ifINDEX == "2":
    channelList = ['36','40', '44', '48'] #5GHz WiFi channels
    #channelList = ['36','40']
    #channelList = ['44', '48']
    #channelList = ['48']
elif ifINDEX == "3":
    channelList = ['1','4', '7', '11'] #TVWS channels
    #channelList = ['1','4']
    #channelList = ['7','11']
    #channelList = ['1']

channelList5G = ['36']
channelListTV = ['8']

#channelWidthList = ['20', '10', '5']
#channelWidthList = ['5','10']
#channelWidthList = ['5']
channelWidthList = ['10']


#txPowerList = ["20", "15", "10", "5"]
#txPowerList = ["20", "15", "10", "5"]
#txPowerList = [ "15"]
txPowerList = ["20"]



pingPacketCount = "20" #number of packets to ping with
iperfTimeout = 150 #seconds be iperf gives up on trying to compute throughput
signal_sample_interval = '1'
signal_sample_duration = ' 20 ' #notice the space before and after value. We choose 20 because iperf runs for 20 seconds (best case scenario)
autossh_forward_port = ' -p 20000 ' #port used to enable 3g control link between A and B via ssh port forwarding
scp_forward_port = ' -P 20000 ' #scp uses capital P
#*********************************************************************
# Files
#*********************************************************************
temp_file_signal_noise = 'temp_signal-noise_samples.log' # file to temporarily keep signal strenght and noise values
temp_file_signal_noise_remote = "temp_signal_samples_remote.log" #when we scp file, we name it this

#********************************************************************
# commands to execute on local node:
#*******************************************************************
cmd_set_txPower = "uci set wireless.@wifi-device["+ ifINDEX +"].txpower="
cmd_set_channel = "uci set wireless.@wifi-device["+ ifINDEX +"].channel="
cmd_set_chanbw = "uci set wireless.@wifi-device["+ ifINDEX +"].chanbw=" #At the moment the nodes are set up such that: wlan3 = TVWS; wlan2 = 5 GHz wifi panel
cmd_uci_commit = "uci commit wireless" #this is all we need to commit, right?
```

```
cmd_reload_network_config = "/etc/init.d/network reload" #reload config file or
    restart network service: etc/init.d/network restart
cmd_get_iw = "iwconfig "+ interface
cmd_get_ifconfig = "ifconfig "+ ipinterface
cmd_get_iw = "iwinfo "+ interface +" info"
cmd_get_delay_and_packet_loss = "ping "+iperfServer +" -c "+ pingPacketCount #
    number of ping packets

# Ending command with "&" puts it in the background, which allowas us to run it in
    the background while iperf computes throughput
# We use a basch script named "testsignal" and the usage:  testsignal [Interval (s)
    ] [Total time(s)] [interface] [outfile]
# The Daeman runtime is INTERVAL x TIME seconds; NUMBER OF SAMPLES = TIME
cmd_get_signal_samples = "./testsignal "+ signal_sample_interval +
    signal_sample_duration  + interface + " "+ temp_file_signal_noise + " &" #run
    in the background
# Make sure the bash script "./testsignal" does not write terminal through STDOUT
    or STDERR i.e. comment out any "echo" statements
# If the bash script writes to stdout or stdrrm the script will not run properly in
     the background

#****************************************************************************
# SOME NOTES about iperf options:
# (i) -f --format specifies format to bring bandwidth number in (https://iperf.fr/
    iperf-doc.php)
#  'k' = Kbits/sec , 'K' = KBytes/sec, 'm' = Mbits/sec,  'M' = MBytes/sec
# (ii) -r measures throughput bi-directionally. By default iperf only measures
    throughput from client to server.
# with the -r option, throughput is measured sequantially i.e. A->B and then A<-B.
# (iii) The -d option would've been ideal because the throughputs A->B and then A<-
    B are measured simultaneously.
# Howver, the -d option is supported in the single threaded iperf version, which is
     what we have currently.
#****************************************************************************
#cmd_get_throughput = "iperf -V -c "+iperfServer +" -f m " #use this flavour with -
    V to support ipv6 for iperf server addresss. This hangs when used with -r
    option.
cmd_get_throughput = "iperf -c "+iperfServer +" -f m -r"




#****************************************************************************
# Commands to execute on remote node:
#****************************************************************************



ssh_password = sshPassword #ssh password for remote node, we got password as
    runtime argument
ssh_newkey = 'Are you sure you want to continue connecting'

#*****************************************************
# When using 3g modem for control then at runtime pass "127.0.0.1" as control link
    ip address
# The ssh and scp syntax are slightly different when going over tunneled link
#**********************************************************************

if controlLink == "127.0.0.1": #this means controlink established using 3g modems
    cmd_set_channel_remote = 'ssh root@'+ controlLink + autossh_forward_port +'uci
    set wireless.@wifi-device['+ remote_ifINDEX +'].channel=' #notice the single
    quotes (') instead of double (")
    cmd_set_txPower_remote = 'ssh root@'+ controlLink + autossh_forward_port +'uci
    set wireless.@wifi-device['+ remote_ifINDEX +'].txpower='
    #get signal strength and noise at remote node
    cmd_get_iw_remote = 'ssh root@'+ controlLink + autossh_forward_port +'iwinfo '+
     remoteInterface +' info' #currently not being used, no longer needed because
```

```
    we're using ./testsignal
    # At the moment the nodes are set up such that: wlan3 = TVWS; wlan2 = 5 GHz wifi panel
    cmd_set_chanbw_remote = 'ssh root@'+ controlLink + autossh_forward_port +'uci set wireless.@wifi-device['+ remote_ifINDEX +'].chanbw=' #use this if measuring WiFi link
    cmd_get_signal_samples_remote = 'ssh root@'+ controlLink + autossh_forward_port +'./testsignal '+ signal_sample_interval + signal_sample_duration+ remoteInterface +' '+ temp_file_signal_noise +' &' #run in the background
    #cmd_delete_remote_temp_file = 'ssh root@'+ controlLink + ' rm '+ temp_file_signal_noise + ' &'
    cmd_delete_remote_temp_file = 'ssh root@'+ controlLink + autossh_forward_port + 'rm temp_signal-noise_samples.log'

else: # else WiFi or TVWS is being used for control purposes
    cmd_set_channel_remote = 'ssh root@'+ controlLink +' uci set wireless.@wifi-device['+ remote_ifINDEX +'].channel=' #notice the single quotes (') instead of double (")
    cmd_set_txPower_remote = 'ssh root@'+ controlLink +' uci set wireless.@wifi-device['+ remote_ifINDEX +'].txpower='
    #get signal strength and noise at remote node
    cmd_get_iw_remote = 'ssh root@'+controlLink +' iwinfo '+ remoteInterface +' info' #currently not being used, no longer needed because we're using ./testsignal
    # At the moment the nodes are set up such that: wlan3 = TVWS; wlan2 = 5 GHz wifi panel
    cmd_set_chanbw_remote = 'ssh root@'+controlLink +' uci set wireless.@wifi-device['+ remote_ifINDEX +'].chanbw=' #use this if measuring WiFi link
    cmd_get_signal_samples_remote = 'ssh root@'+ controlLink + ' ./testsignal '+ signal_sample_interval + signal_sample_duration+ remoteInterface +' '+ temp_file_signal_noise +' &' #run in the background
    #cmd_delete_remote_temp_file = 'ssh root@'+ controlLink + ' rm '+ temp_file_signal_noise + ' &'
    cmd_delete_remote_temp_file = 'ssh root@'+ controlLink + ' rm temp_signal-noise_samples.log'



#*************************************************************
# Timeout() class is aimed at limiting the amount of time iperf spends
# trying to compute throughout. This is particularly an issue when the
# the link is bad. The -t option that is supposed to be used to specify
# iperf duration does is buggy. The current iperf implementation
# returns the output only after the entire block is sent.
#*******************************************************************
# I got the trick from https://pythonadventures.wordpress.com/2012/12/08/raise-a-timeout-exception-after-x-seconds/
class Timeout():
    """Timeout class using ALARM signal."""
    class Timeout(Exception):
        pass

    def __init__(self, sec):
        self.sec = sec

    def __enter__(self):
        signal.signal(signal.SIGALRM, self.raise_timeout)
        signal.alarm(self.sec)

    def __exit__(self, *args):
        signal.alarm(0)    # disable alarm

    def raise_timeout(self, *args):
        raise Timeout.Timeout()

#End Timeout()

```

```python
#**********************************************************************
# standard_deviation computes the standard deviation of a list of values
# We've particularly used it to compute the standard deviation of multple signal
    and noise values
# Credit should go to Josh (https://codeselfstudy.com/blogs/how-to-calculate-
    standard-deviation-in-python)
#*************************************************************************
def standard_deviation(lst, population=True):
    """Calculates the standard deviation for a list of numbers."""
    num_items = len(lst)
    mean = sum(lst) / float(num_items) #convert one value to float, otherwise it's
    integer division
    differences = [x - mean for x in lst]
    sq_differences = [d ** 2 for d in differences]
    ssd = sum(sq_differences)

    # Note: it would be better to return a value and then print it outside
    # the function, but this is just a quick way to print out the values along
    # the way.
    if population is True:
        #print('This is POPULATION standard deviation.')
        variance = ssd / num_items
    else:
        #print('This is SAMPLE standard deviation.')
        variance = ssd / (num_items - 1)
    sd = math.sqrt(variance)
    # You could `return sd` here.

    #print('The mean of {} is {}.'.format(lst, mean))
    #print('The differences are {}.'.format(differences))
    #print('The sum of squared differences is {}.'.format(ssd))
    #print('The variance is {}.'.format(variance))
    #print('The standard deviation is {}.'.format(sd))
    #print('--------------------------')
    return sd
#***End standard_deviation

#**********************************************************************
# Main measurement loop starts here
#**********************************************************************

# Lock the channel,power,channel width set for radio




with open(filename, "w") as outfile:
#csvWriter = csv.writer(outfile, delimiter = ',', quoting = csv.QUOTE_MINIMAL)
    csvWriter = csv.writer(outfile, delimiter = ',', lineterminator = '\n')

    readings = [] #create "table" or structure to add stuff to

    #label the columns
    header = ["Channel no.","Channel-width (MHz)","TxPower (dBm)", "A->B Throughput
     (Mbits/sec)", "A<-B Throughput (Mbits/sec)", "BitRate (MBits/s)", "(A) Min
    SignalStrength (dBm)", "(A) Mean SignalStrength (dBm)", "(A) Max SignalStrength
     (dBm)", "(A) STDEV signal", "(A) Min Noise (dBm)",  "(A) Mean Noise (dBm)", "(
    A) Max Noise (dBm)","(A) STDEV noise", "(B) Min SignalStrength (dBm)", "(B)
    Mean SignalStrength (dBm)", "(B) Max SignalStrength (dBm)", "(B) STDEV signal",
     "(B) Min Noise (dBm)",  "(B) Mean Noise (dBm)", "(B) Max Noise (dBm)","(B)
    STDEV noise", "TxPackets", "TxErrors", "RxPackets", "RxErrors", "packet loss
    (%)", "RTT_min (ms)", "RTT_avg (ms)", "RTT_max (ms)"]
    csvWriter.writerow(header)
    C = 0 #the conventional 'i' is being used for other things so, improvising with
     'C'
    for eachChannel in channelList5G:

```

```
        #set the channel
        channel5G = channelList5G[C]
        channelTV = channelListTV[C]
        C = C + 1

        #set the channel on remote host and apply settings
        print "setting the WiFi channel on remote machine ["+ iperfServer +"]: "+
    cmd_set_channel + channel5G
        remote_ifINDEX = "1"
        p=pexpect.spawn(cmd_set_channel_remote + channel5G + '&& uci commit
    wireless; wifi &') #the & at end of command forces exist after issuing command

        i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==0:
            p.sendline('yes') # say yes to the question "areyou sure you want to
    continue connecting?"
            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==1:
            p.sendline(ssh_password) #enter ssh password
            p.expect(pexpect.EOF)
        elif i==2:
            pass # either connection is successful or failed for some reason
        print p.before # print out the result

        print "setting the TVWS channel on remote machine ["+ iperfServer +"]: "+
    cmd_set_channel + channelTV
        remote_ifINDEX = "3"
        p=pexpect.spawn(cmd_set_channel_remote + channelTV + '&& uci commit
    wireless; wifi &') #the & at end of command forces exist after issuing command

        i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==0:
            p.sendline('yes') # say yes to the question "areyou sure you want to
    continue connecting?"
            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==1:
            p.sendline(ssh_password) #enter ssh password
            p.expect(pexpect.EOF)
        elif i==2:
            pass # either connection is successful or failed for some reason
        print p.before # print out the result



        #time.sleep(5)

        #set the channel on localhost
        ifINDEX = "1"
        print "Setting the WiFi channel on localhost: "+ cmd_set_channel +
    channel5G
        cmd = subprocess.Popen(cmd_set_channel + channel5G + '&& uci commit
    wireless; wifi', shell = True, stdout = subprocess.PIPE, stderr = subprocess.
    STDOUT)
        #time.sleep(60)

        ifINDEX = "3"
        print "Setting the TV channel on localhost: "+ cmd_set_channel + channelTV
        cmd = subprocess.Popen(cmd_set_channel + channelTV + '&& uci commit
    wireless; wifi', shell = True, stdout = subprocess.PIPE, stderr = subprocess.
    STDOUT)

        print cmd.stdout.read() #display errors if any, otherwise output=<empty
        time.sleep(15)


        #set channel bandwith
        k = 0
```

```
    for eachChannelWidth in channelWidthList:
        chanbw = channelWidthList[k]
        k = k + 1

        remote_ifINDEX = "1"
        #set channel width on remote host
        print "Setting the channel width for WiFi on remote machine: "+ cmd_set_chanbw_remote + chanbw
        #Run the commond "uci commit wireless" &&...
        p=pexpect.spawn(cmd_set_chanbw_remote + chanbw +' && uci commit wireless && /etc/init.d/network reload &')
        #p=pexpect.spawn('ssh root@10.1.5.50 uci set wireless.@wifi-device[2].chanbw=20 && uci commit wireless && /etc/init.d/network reload && exit')

        i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==0:
            p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==1:
            p.sendline(ssh_password) #enter ssh password
            p.expect(pexpect.EOF)
        elif i==2:
            pass # either connection is successful or failed for some reason
        print p.before # print out the result

        remote_ifINDEX = "3"
        #set channel width on remote host
        print "Setting the channel width for TVWS on remote machine: "+ cmd_set_chanbw_remote + chanbw
        #Run the commond "uci commit wireless" &&...
        p=pexpect.spawn(cmd_set_chanbw_remote + chanbw +' && uci commit wireless && /etc/init.d/network reload &')
        #p=pexpect.spawn('ssh root@10.1.5.50 uci set wireless.@wifi-device[2].chanbw=20 && uci commit wireless && /etc/init.d/network reload && exit')

        i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==0:
            p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
        if i==1:
            p.sendline(ssh_password) #enter ssh password
            p.expect(pexpect.EOF)
        elif i==2:
            pass # either connection is successful or failed for some reason
        print p.before # print out the result




        #set channel width for WiFi on localhost
        ifINDEX = "1"
        print "Setting the channel width on localhost: "+cmd_set_chanbw + chanbw
        cmd = subprocess.Popen(cmd_set_chanbw + chanbw, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
        #time.sleep(60)
        print cmd.stdout.read() #display errors if any, otherwise output=<empty
        cmd = subprocess.Popen(cmd_uci_commit, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
        #time.sleep(60)
        print cmd.stdout.read() #display errors if any, otherwise output=<empty
        time.sleep(5)

        #set channel width for TVWS on localhost
        ifINDEX = "3"
```

```
        print "Setting the channel width on localhost: "+cmd_set_chanbw + chanbw
        cmd = subprocess.Popen(cmd_set_chanbw + chanbw, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
        #time.sleep(60)
        print cmd.stdout.read() #display errors if any, otherwise output=<empty
        cmd = subprocess.Popen(cmd_uci_commit, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
        #time.sleep(60)
        print cmd.stdout.read() #display errors if any, otherwise output=<empty
        time.sleep(5)


        j = 0
        for eachTxPower in txPowerList:

            #set tx power
            txPower = txPowerList[j]
            j = j + 1

            remote_ifINDEX = "1"

            print "setting the Tx-power for WiFi on remote node: "+cmd_set_txPower_remote + txPower
            #Run the commond "uci commit wireless" &&...
            p=pexpect.spawn(cmd_set_txPower_remote + txPower +' && uci commit wireless && /etc/init.d/network reload &')
            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
            if i==0:
                p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
                i=p.expect([ssh_newkey,'password:',pexpect.EOF])
            if i==1:
                p.sendline(ssh_password) #enter ssh password
                p.expect(pexpect.EOF)
            elif i==2:
                pass # either connection is successful or failed for some reason
            print p.before # print out the result

            remote_ifINDEX = "3"

            print "setting the Tx-power for TVWS on remote node: "+cmd_set_txPower_remote + txPower
            #Run the commond "uci commit wireless" &&...
            p=pexpect.spawn(cmd_set_txPower_remote + txPower +' && uci commit wireless && /etc/init.d/network reload &')
            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
            if i==0:
                p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
                i=p.expect([ssh_newkey,'password:',pexpect.EOF])
            if i==1:
                p.sendline(ssh_password) #enter ssh password
                p.expect(pexpect.EOF)
            elif i==2:
                pass # either connection is successful or failed for some reason
            print p.before # print out the result


            ifINDEX = "1"
            print "Setting the Tx-power for WiFi on local node: "+cmd_set_txPower + txPower
            cmd = subprocess.Popen(cmd_set_txPower + txPower +" && "+cmd_uci_commit, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
```

```
            #time.sleep(60)
            print cmd.stdout.read() #display errors if any, otherwise output=<empty>

            ifINDEX = "3"
            print "Setting the Tx-power for TVWS on local node: "+cmd_set_txPower + txPower
            cmd = subprocess.Popen(cmd_set_txPower + txPower +" && "+cmd_uci_commit, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
            #time.sleep(60)
            print cmd.stdout.read() #display errors if any, otherwise output=<empty>



            #reload network config
            print "Applying network settings on local node: "+cmd_reload_network_config
            cmd = subprocess.Popen(cmd_reload_network_config, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
            time.sleep(5)
            print cmd.stdout.read() #display errors if any, otherwise output=<empty>


            #Rebuild batman channel bonding maybe ?


            #note txpower and channel in use
            #run the iwconfig command to get bitRate, signal strength, noise

            print "Executing "+cmd_get_iw
            cmd = subprocess.Popen(cmd_get_iw, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)

            output = cmd.stdout.read()
            #see https://docs.python.org/2/howto/regex.html
            #uncomment lines below on Ubuntu
            #bitRate = re.findall(r'Bit Rate=(\d*.\d*)', output)
            #signalStrength = re.findall(r'Signal level=(\D\d*)', output)
            #noise = re.findall(r'Noise (\d*)', output) #refine this for ubuntu
            #on openwrt:

            # V5 no longer records signal and noise this way
            bitRate = re.findall(r'Bit Rate: (\d*.\d*)', output)
            signalStrength = re.findall(r'Signal: (\D\d*)', output)
            noise = re.findall(r'Noise: (\D\d*)', output)

            #if no value found, write not any ("N/A") in the appropriate column
            if not bitRate:
                bitRate = ["n/a"]
            if not signalStrength:
                signalStrength = ["n/a"]
            if not noise:
                noise = ["n/a"]

            print output #display iwconfig output, non real-time:(

            #**********************************
            # Attemping to get signal and noise on remote node
            # This was before we developed scheme to get multiple sample and compute min/mean/max/stdev
            #*********************************************************
            '''
            print "Attempting to get signal strength and noise of remote node.."
```

```
            # uci commit is not necessary, but's it's the only way I've been able to hook the remote output
            p=pexpect.spawn(cmd_get_iw_remote + '&& uci commit wireless; wifi &') #the & at end of command forces exist after issuing command

            i=p.expect([ssh_newkey,'password:',pexpect.EOF])
            if i==0:
                p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
                i=p.expect([ssh_newkey,'password:',pexpect.EOF])
            if i==1:
                p.sendline(ssh_password) #enter ssh password
                p.expect(pexpect.EOF)
            elif i==2:
                pass # either connection is successful or failed for some reason
            print p.before # print out the result
            signalStrengthB = re.findall(r'Signal: (\D\d*)', p.before)
            noiseB = re.findall(r'Noise: (\D\d*)', p.before)

            if not signalStrengthB:
                signalStrengthB =["n/a"]
            if not noiseB:
                noiseB = ["n/a"]
            '''
            #End attempt to get signal and noise of remote node

            #run ifconfig to get a sense of packet error rate
            print "Executing "+ cmd_get_ifconfig
            cmd = subprocess.Popen(cmd_get_ifconfig, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)

            output = cmd.stdout.read()

            rxPackets = re.findall(r'RX packets:(\d*)', output)
            rxErrors= re.findall(r'RX packets:\d* errors:(\d*)', output)
            txPackets = re.findall(r'TX packets:(\d*)', output)
            txErrors = re.findall(r'TX packets:\d* errors:(\d*)', output)

            #if no value found, write not any ("N/A") in the appropriate column
            if not rxPackets:
                rxPackets = ["n/a"]
            if not rxErrors:
                rxErrors = ["n/a"]
            if not txPackets:
                txPackets = ["n/a"]
            if not rxErrors:
                txErrors = ["n/a"]

            print output #display ifconfig output
            #time.sleep(15)

            # ******************************************************************************

            # Just before launching iperf, get multiple samples of signal and noise values for the local node
            # This subprocess runs in the background and the idea is to take the samples while iperf computes the throughput
            print "Attempting to start signal strength and noise sampling at remote node: "+ cmd_get_signal_samples_remote
            # uci commit is not necessary, but's it's the only way I've been able to hook the remote output
            #p=pexpect.spawn(cmd_get_iw_remote + '&& uci commit wireless; wifi &') #the & at end of command forces exist after issuing command
            p=pexpect.spawn(cmd_get_signal_samples_remote)  #leave process running in the background and exit
```

```
                i=p.expect([ssh_newkey,'password:',pexpect.EOF])
                if i==0:
                    p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
                    i=p.expect([ssh_newkey,'password:',pexpect.EOF])
                if i==1:
                    p.sendline(ssh_password) #enter ssh password
                    p.expect(pexpect.EOF)
                elif i==2:
                    pass # either connection is successful or failed for some reason               #

                #print p.before
                #Fire up testsignal daemon at localhost
                cmd_D = subprocess.Popen(cmd_get_signal_samples, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
                print "Started testsignal subprocess on localhost: ", cmd_get_signal_samples, "..."

                try:
                    with Timeout(iperfTimeout):  #iperf will timeout after this amount of time if not complete by then
                        #run iperf command to get throughput
                        #print "[channel="+channel+"]"+"[chnnelWidth="+chanbw+"]" + "[txpower="+txPower+"]"
                        print "\n Current settings: channel="+channel+", "+"chnnelWidth="+chanbw+", " + "txpower="+txPower
                        print "Running iperf: "+ cmd_get_throughput
                        cmd = subprocess.Popen(cmd_get_throughput, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
                        output = cmd.stdout.read()

                except Timeout.Timeout:
                    print "iperf timeout out after ", iperfTimeout, "seconds"

                #"findall" will find all matches and return them as a list
                # throughput[0] implies first match, which is A to B
                # throughput[1] implies second match, which is B to A
                throughput = re.findall(r'MBytes  (\d*.\d*)', output)
                #try getting value just before "Mbits/sec" instead of after "MBytes" we've locked output format to Mbits/sec
                # *********************************************************************
                # This part needs some work, when geting throughput in both directions.
                # currents the script bombs out if there's only throughput in one direction and not the other
                # ****************************************************************************
                #if no value found, write not any ("N/A") in the Throughput column
                if not throughput:
                    throughput= ["n/a", "n/a"] #no throughput recorded from A to B
                elif len(throughput)== 1:
                    throughput.append ("n/a") #no throughput recored from B to A.
                    #This needs to be worked on because it could be that the one element in the list is actually B to A
                    #As it is, it's only acurate if A to B is recorded and B to A is not recorded
                print output #display iperf output, non real-time:(

                #****************************************************************
                # Get multiple signal and noise samples
                # By this time iperf completes,testsignal should also be done
                # **************************************************************
                # transfer file created by testsignal from remote node to local node

```

```
                print "Fetching signal and noise samples from remote node..."
                if controlLink == "127.0.0.1":
                    p=pexpect.spawn('scp'+ scp_forward_port + 'root@'+controlLink+':'+temp_file_signal_noise + ' '+temp_file_signal_noise_remote)
                else:
                    p=pexpect.spawn('scp root@'+controlLink+':'+temp_file_signal_noise + ' '+temp_file_signal_noise_remote)
                i=p.expect([ssh_newkey,'password:',pexpect.EOF])
                if i==0:
                    p.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
                    i=p.expect([ssh_newkey,'password:',pexpect.EOF])
                if i==1:
                    p.sendline(ssh_password) #enter ssh password
                    p.expect(pexpect.EOF)
                elif i==2:
                    pass # either connection is successful or failed for some reason

                #print p.before

                print "Signal and noise samples from remote node:"
                signal_samples_remote = []
                noise_samples_remote = []
                try:
                    with open(temp_file_signal_noise_remote) as rf:
                        for line in rf:
                            data_remote = line.split() #split on the space
                            if not data_remote: #this prevents "list index out of range" error, which is cause by reading past eof
                                #added this code in a rush 16 Nov 2017
                                min_signal_remote = "n/a"
                                mean_signal_remote = "n/a"
                                max_signal_remote = "n/a"
                                stdev_signal_remote = "n/a"
                                min_noise_remote = "n/a"
                                mean_noise_remote = "n/a"
                                max_noise_remote = "n/a"
                                stdev_noise_remote = "n/a"
                                break
                            else:
                                try: #prevent script from bombing out if file contains "unknown dBm" text instead of numbers
                                    signal_samples_remote.append(int(data_remote[0]))
                                    noise_samples_remote.append(int(data_remote[1]))
                                except ValueError:
                                    continue
                                print data_remote[0] , ' ', data_remote[1]
                except EnvironmentError:
                    print 'Problem fetching signal and noise samples from remote host'
                if not signal_samples_remote:
                    min_signal_remote = "n/a"
                    mean_signal_remote = "n/a"
                    max_signal_remote = "n/a"
                    stdev_signal_remote = "n/a"
                else:
                    min_signal_remote = min(signal_samples_remote)
                    mean_signal_remote = float(sum(signal_samples_remote))/float(len(signal_samples_remote))
                    max_signal_remote = max(signal_samples_remote)
                    if len(signal_samples_remote) == 1: #prevent divisio by zero in standrd_deviation()
                        stdev_signal_remote = "n/a"
                    else:
```

```
                    stdev_signal_remote = standard_deviation(signal_samples_remote, population=False) #sample standard deviation instead of population standard deviation
            if not noise_samples_remote:
                min_noise_remote = "n/a"
                mean_noise_remote = "n/a"
                max_noise_remote = "n/a"
                stdev_noise_remote = "n/a"
            else:
                min_noise_remote = min(noise_samples_remote)
                mean_noise_remote = float(sum(noise_samples_remote))/float(len(noise_samples_remote))
                max_noise_remote = max(noise_samples_remote)
                if len(noise_samples_remote)==1:
                    stdev_noise_remote = "n/a" ##prevent divisio by zero in standrd_deviation()
                else:
                    stdev_noise_remote = standard_deviation(noise_samples_remote, population=False) #sample standard deviation
            print "\n _____ Node B _____"
            print "\t Min signal: ", min_signal_remote
            print "\t Mean signal: ", mean_signal_remote
            print "\t Max signal: ", max_signal_remote
            print "\t STDEV signal: ", stdev_signal_remote
            print "\t Min noise: ", min_noise_remote
            print "\t Mean noise: ", mean_noise_remote
            print "\t Max noise: ", max_noise_remote
            print "\t STDEV noise: ", stdev_noise_remote

            print "\n Signal and noise samples at local node:"
            signal_samples = []
            noise_samples = []

            with open (temp_file_signal_noise) as f:  #read the contents of file created by cmd_D, assuming everything went ok.
                for line in f:
                    data = line.split( ) #split on the space
                    if not data: #this prevents "list index out of range" error, which is cause by reading past eof
                        #Added this code in a rush 16 NOV 2017
                        min_signal = "n/a"
                        mean_signal = "n/a"
                        max_signal = "n/a"
                        stdev_signal = "n/a"
                        min_noise = "n/a"
                        mean_noise = "n/a"
                        max_noise = "n/a"
                        stdev_noise = "n/a"
                        break
                    else:
                        try: #prevent script from bombing out if file contains "unknown dBm" text instead of numbers
                            signal_samples.append(int(data[0]))
                            noise_samples.append(int(data[1]))
                        except ValueError:
                            continue
                        print data[0], ' ', data[1]
            if not signal_samples:
                min_signal = "n/a"
                mean_signal = "n/a"
                max_signal = "n/a"
                stdev_signal = "n/a"
            else:
                min_signal = min(signal_samples)
                mean_signal = float(sum(signal_samples))/float(len(signal_samples))
                max_signal = max(signal_samples)
```

```
                if len(signal_samples)==1:
                    stdev_signal = "n/a"
                else:
                    stdev_signal = standard_deviation(signal_samples, population=False) #sample standard deviation instead of population standard deviation
            if not noise_samples:
                min_noise = "n/a"
                mean_noise = "n/a"
                max_noise = "n/a"
                stdev_noise = "n/a"
            else:
                min_noise = min(noise_samples)
                mean_noise = float(sum(noise_samples))/float(len(noise_samples))
                max_noise = max(noise_samples)
                if len(noise_samples)==1:
                    stdev_noise = "n/a"
                else:
                    stdev_noise = standard_deviation(noise_samples, population=False) #sample standard deviation
            print "\n_____ Node A _____"
            print "\t Min signal: ", min_signal
            print "\t Mean signal: ", mean_signal
            print "\t Max signal: ", max_signal
            print "\t STDEV signal: ", stdev_signal
            print "\t Min noise: ", min_noise
            print "\t Mean noise: ", mean_noise
            print "\t Max noise: ", max_noise
            print "\t STDEV noise: ", stdev_noise

            #***end multiple signal strength values

            print "\n Current settings: channel="+channel+", "+"chnnelWidth="+chanbw+", " + "txpower="+txPower
            print "Running ping: "+cmd_get_delay_and_packet_loss
            cmd = subprocess.Popen(cmd_get_delay_and_packet_loss, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
            output = cmd.stdout.read()
            #****************************************
            #Get packet loss, rtt min/avg/max
            #*************************************************

            #get packet loss, which is value after the word 'received,' in the ping output
            packet_loss = re.findall(r'received, (\d*)', output)
             #get the min RRT, which is the value right after the expression "min/avg/max = "
            rtt_min = re.findall(r'min/avg/max = (\d*.\d*)', output)
            if not rtt_min:
                rtt_min = ["n/a"]
            #get the avg RRT, which is the value right after the expression "min/avg/max = xx.xx/"
            rtt_avg =  re.findall(r'min/avg/max = \d*.\d*/(\d*.\d*)', output)
            if not rtt_avg:
                rtt_avg = ["n/a"]
             #get the max RRT, which is the value right after the expression "min/avg/max = xx.xx/xx.xx/"
            rtt_max = re.findall(r'min/avg/max = \d*.\d*/\d*.\d*/(\d*.\d*)', output)
            if not rtt_max:
                rtt_max = ["n/a"]
            print output #display ping output


            #append values to readings
```

```
                # throughput, bitrate, signalStrength.....RxErros are lists. Therefore, use [0] to get value and leave out brackets when writing
                # to csv file. There's only one element in the list so, [0] is adequate, except for throughput where we have two
                # 'findall' finds all matches and returns them as a list.
                # throughput[0] = throughput of A to B, throughput[1] = throughput of B to A
                readings = [channel, chanbw, txPower, throughput[0], throughput[1], bitRate[0], min_signal, mean_signal, max_signal, stdev_signal, min_noise, mean_noise, max_noise, stdev_noise, min_signal_remote, mean_signal_remote, max_signal_remote, stdev_signal_remote, min_noise_remote, mean_noise_remote, max_noise_remote, stdev_noise_remote, txPackets[0], txErrors[0], rxPackets[0], rxErrors[0], packet_loss[0], rtt_min[0], rtt_avg[0], rtt_max[0]]#concatenate list and str elements
                csvWriter.writerow(readings) #write row to file
                print "Output written to "+filename+"\n"
                #*************************************************
                # Perform some housekeeping:
                # delete the temp files with signal and noise samples.
                # In case something goes wrong with ./testsignal, we don't to pull old values from the files.
                # There, we delete the files and create them afresh on each iteration
                # ************************************************
                print "_________Cleaning up_________"
                print "\t Node A: Deleting "+ temp_file_signal_noise_remote #this is the file scp'd from remote host
                cmd = subprocess.Popen('rm '+temp_file_signal_noise_remote, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
                print cmd.stdout.read() #display errors if any, otherwise output=< empty
                print "\t Node A: Deleting "+ temp_file_signal_noise #this is the fie created by ./testsignal on the local node
                cmd = subprocess.Popen('rm '+temp_file_signal_noise, shell = True, stdout = subprocess.PIPE, stderr = subprocess.STDOUT)
                print cmd.stdout.read() #display errors if any, otherwise output=< empty

                print "\t Node B: Deleting "+ temp_file_signal_noise #delete the file created by ./testsignal on remote node
                p2=pexpect.spawn(cmd_delete_remote_temp_file )
                i=p2.expect([ssh_newkey,'password:',pexpect.EOF])
                if i==0:
                    p2.sendline('yes') # say yes to the question "areyou sure you want to continue connecting?"
                    i=p2.expect([ssh_newkey,'password:',pexpect.EOF])
                if i==1:
                    p2.sendline(ssh_password) #enter ssh password
                    p2.expect(pexpect.EOF)
                elif i==2:
                    pass # either connection is successful or failed for some reason
                print p2.before

outfile.close()
print "done"

#<delete temporal file created for signal samples>?
```

Listing D.1: Python master measurement script.

```
#!/bin/sh

ip link set mtu 1532 dev wlan1-mesh
ip link set mtu 1532 dev wlan3-mesh
batctl if add wlan1-mesh
batctl if add wlan3-mesh
ip link set up dev wlan1-mesh
ip link set up dev wlan3-mesh
ip link set up dev bat0
ip addr add 10.1.7.50/24 dev bat0
batctl bonding enable
```

Figure D.1: Shell script used to bond WiFi and TVWS Links.

```
#!/bin/sh

if [ "$#" -ne 4 ]; then
    echo "Usage: testsignal [Interval (s)] [Total time(s)] [interface] [file name]"
    exit
fi

interval=$1
time=$2
iface=$3
fname=$4

echo $interval
echo $iface

for i in `seq 1 $time`;
do
    iwinfo $iface info | grep Signal | awk -F " " '{print $2,$5}' >> $fname
    sleep $interval
done
```

Figure D.2: Signal strength testing script.

APPENDIX E

TABULATED THROUGHPUT VALUES OF SINGLE AND AGGREGATED WIFI AND TVWS LINKS FROM, DOWNLINK TO UPLINK AND UPLINK TO DOWNLINK PATH

WiFi	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**TVWS**	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**Aggregated** Mbps
36	10	11.42	-62.5	-106	8	10	10.26	-39.7	-97	16.28
36	10	11.42	-72.3	-106	8	5	3.05	-44.5	-104	12.56
36	10	11.58	-75.6	-106	8	20	14.1	-12.8	-97	18.08
36	5	5.80	-79.3	-109	8	20	13.94	-16.0	-96	12.56
36	5	4.61	-82.0	-109	8	10	8.58	-14.4	-101	7.03
36	5	2.82	-93.4	-109	8	5	3.17	-15.6	-104	3.54
36	20	11.94	-87.7	-103	8	5	3.12	-15.4	-104	8.99
36	20	8.11	-87.8	-103	8	10	8.56	-12.6	-107	13.94
36	20	13.58	-84.9	-103	8	20	13.76	-13.5	-97	19.16
WiFi	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**TVWS**	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**Aggregated** Mbps
44	20	22.06	-67.3	-103	1	20	12.7	-32.5	-95	22.98
44	20	22.86	-67.2	-103	1	10	4.37	-18.7	-98	9.96
44	20	22.86	-67.5	-103	1	5	2.31	-35.9	-102	23.96
44	10	12.00	-67.2	-106	1	5	2.14	-15.2	-102	8.02
44	10	12.00	-67.2	-106	1	10	4.57	-66.3	-106	12.28
44	10	12.08	-66.5	-106	1	20	13.62	-12.6	-95	16.62
44	5	6.15	-68.1	-109	1	20	14.38	-14.6	-95	13.32
44	5	6.19	-67.6	-109	1	10	3.82	-32.9	-98	6.65
44	5	6.02	-68.3	-109	1	5	2.51	-11.8	-100	4.19
WiFi	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**TVWS**	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**Aggregated** Mbps
48	5	5.78	-68.6	-109	11	5	3.71	-14.1	-100	6.39
48	5	5.92	-68.3	-108	11	10	8.77	-14.1	-97	9.58
48	5	5.84	-69.2	-109	11	20	9.05	-41.3	-91	8.77
48	10	12.02	-67.6	-106	11	20	-13.6	-92	16.12	
48	10	12.00	-68.2	-106	11	10	8.85	-41.2	-97	14.64
48	10	11.98	-66.7	-106	11	5	3.94	-14.0	-100	7.34
48	20	22.68	-67.7	-103	11	5	3.99	-14.4	-101	7.41
48	20	22.66	-67.6	-103	11	10	7.61	-13.4	-97	18.72
48	20	22.76	-66.6	-103	11	20	12.68	-41.6	-91	15.84

Table E.1: Aggregated WiFi-TVWS Link throughput at possible channel and channel width link permutations (From rooftop node to ground node).

WiFi	ChanWidth MHz	Throughput Mbps	Signal dBm	Noise dBm	TVWS	ChanWidth MHz	Throughput Mbps	Signal dBm	Noise dBm	Aggregated Mbps
36	10	11.48	-57.4	-106	8	10	10.34	-39	-92.6	16.96
36	10	10.64	-69.7	-105	8	5	4.16	-39.3	-94	error
36	10	11.34	-71.1	-105	8	20	18.500	-11.1	-86	18.44
36	5	5.64	-72	-108	8	20	0.17	-39.19	-87	14.76
36	5	3.51	-80.6	-108	8	10	10.22	-39.2	-92	8.92
36	5	3.56	-92	-108	8	5	3.714	-15.6	-95	4.98
36	20	10.65	-81	-102	8	5	3.93	-39.1	-96	7.48
36	20	8.11.37	-81	-102	8	10	10.3	-9.7	-92	16.98
36	20	12.98	-75.6	-102	8	20	16.4	-39.8	-87.6	17.14
WiFi	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**TVWS**	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**Aggregated** Mbps
44	20	22.46	-62.3	-103	1	20	9.24	-34.9	-86	14.7
44	20	22.6	-63.8	-103	1	10	7.70	-34.8	-92	14.26
44	20	22.64	-63.5	-103	1	5	3.39	-36.8	-101	6.85
44	10	11.02	-63.9	-106	1	5	3.31	-35.4	-95	8.95
44	10	11.9	-63.9	-106	1	10	7.62	-34.8	-91	15.02
44	10	11.8	-62.8	-105.2	1	20	17.98	-64.2	-105	19.46
44	5	5.99	-64.4	-108	1	20	19	-35.3	-83	18.1
44	5	5.93	-64.6	-108	1	10	7.7	-34.3	-89	9.18
44	5	3.78	-63.2	-108	1	5	3.36	-34	-93	4.72
WiFi	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**TVWS**	**ChanWidth** MHz	**Throughput** Mbps	**Signal** dBm	**Noise** dBm	**Aggregated** Mbps
48	5	2.47	-63.9	-106	11	5	3.67	-41.1	-100	2.99
48	5	1.22	-64	-106	11	10	9.20	-40.6	-93	3.30
48	5	4.38	-64	-108	11	20	17.7	-40.6	-94	10.24
48	10	11.94	-64	-105	11	20	15.46	-41.2	-88	10.24
48	10	11.06	-64.6	-105	11	10	9.05	-40.2	-95	18.16
48	10	10.44	-64.1	-105	11	5	3.94	-41.9	-101	14.52
48	20	22.42	-64.4	-102	11	5	3.89	-41.8	-100	6.68
48	20	22.42	-64.2	-103	11	10	8.98	-41	-95	16.52
48	20	19.26	-62	-103	11	20	9.84	-93	-89	14.42

Table E.2: Aggregated WiFi-TVWS Link throughput at possible channel and channel width link permutations (From ground node to rooftop node).